Институт проблем информационной безопасности Московского государственного университета имени М. В. Ломоносова

Академия криптографии Российской Федерации

Математика и безопасность информационных технологий

Материалы конференции в МГУ 23–24 октября 2003 г.

Содержание

Часть I
Пленарные доклады

Проблемы информационной безопасности в современном мире

В. П. Шерстюк

Государственная политика в области обеспечения безопасности представляет собой целенаправленную деятельность по разработке и реализации мер политического, организационного, правового, экономического, социального, научного и иного характера, направленных на оптимизацию всей системы обеспечения национальной безопасности. Все этапы подготовки и принятия решений в сфере национальной безопасности — от достоверного прогнозирования и анализа угроз до выработки системы адекватных мер для их парирования — одинаково важны.

В наступившем третьем тысячелетии человечество столкнулось с новыми вызовами и угрозами безопасности. На рубеже веков сама система мироустройства претерпевает радикальные структурные изменения. Происходит укрепление экономических и политических позиций целого ряда государств и их интеграционных объединений. На авансцену международной жизни выходят наднациональные структуры, интересы которых по ряду вопросов вступают во взаимные противоречия, противопоставляются интересам отдельных государств. Появилась угроза международного терроризма.

До недавнего времени под национальной безопасностью понималось сохранение суверенитета и территориальной целостности государства, его устойчивость перед угрозой применения вооруженной силы со стороны других субъектов международных отношений. Однако вызовы последнего десятилетия потребовали иных подходов к оценке содержания национальной безопасности. Сегодня национальная безопасность видится как комплексная системная проблема. Она должна рассматриваться в более широком контексте и учитывать наличие многообразных факторов и угроз, а не только угрозы военного нападения, захвата территории и физического уничтожения населения. Это обстоятельство требует проведения глубоких научных исследований фундаментального характера в области национальной безопасности.

В Концепции национальной безопасности (Указ Президента Российской Федерации от 10 января 2000 г.) подчеркивается ведущая роль науки в жизни страны и в решении задач национальной безопасности. Основополагающая роль науки не может быть подменена организационными и оперативными мероприятиями по обеспечению безопасности. Всестороннюю и глубокую научную проработку решений Президента Российской Федерации по наиболее актуальным вопросам обеспечения жизненно важных интересов личности, общества и государства осуществляет Научный совет при Совете безопасности Российской Федерации. Создание подобного органа в системе высшей государственной власти — шаг беспрецедентный для России.

Деятельность Научного совета постоянно нацелена на формирование и реализацию методологии научного обоснования стратегии национальной безопасности в условиях действия разнонаправленных векторов угроз. Работа базируется на потенциале входящих в его состав видных ученых, представителей научной общественности и специалистов оборонно-промышленного комплекса, ректоров учебных заведений, руководящих работников министерств и ведомств. Значительную роль в работе научного совета играют представители Российской академии наук и других академий, имеющих государственный статус, представители Высшей школы.

Значение методологического блока исследования проблем национальной безопасности трудно переоценить. Основным его содержанием является выявление, оценка и прогнозирование угроз национальной безопасности, оценка состояния защищенности национальных интересов и эффективности проводимых мероприятий по обеспечению национальной безопасности.

В последнее время набирают силу процессы научно-методологического обоснования политики Российской Федерации в различных ключевых сферах обеспечения национальной безопасности. Подготовлены и утверждены Военная доктрина и Концепция национальной безопасности Российской Федерации. В развитие Концепции национальной безопасности силы аппарата Совета Безопасности, ведущих научных организаций и ученых научного сообщества, Правительства Российской Федерации, федеральных органов исполнительной власти, Госсовета, федеральных округов, Научного совета

при Совете Безопасности были сосредоточены на разработке целого ряда доктринальных и концептуальных документов. Они становятся ее логическим продолжением.

В совокупности эти документы образуют взаимоувязанную систему взглядов на обеспечение национальной безопасности Российской Федерации. Эта работа ведется непрерывно. Перечень базовых документов постоянно наращивается. Только за последниме неполные 3 года разработаны: Доктрина информационной безопасности Российской Федерации, Концепция внешней политики Российской Федерации, Государственная стратегия экономической безопасности Российской Федерации, Основы политики Российской Федерации в области развития оборонно-промышленного комплекса, в области развития науки и технологий, Основы военно-технической политики, Основы государственной политики в ряде таких важнейших областей деятельности государства, как морская, авиационная и космическая и др. В настоящее время ведется работа по подготовке концептуальных документов в сфере промышленной, научно-технологической и инновационной, химической и биологической безопасности, а также в области обеспечения защищенности особо опасных и критически важных объектов инфраструктуры государства и населения страны. Продолжается работа по развитию и углублению отдельных положений Доктрины информационной безопасности, поскольку одним из наиболее важных факторов, определяющих развитие современного общества, является продолжающаяся «информационная революция».

Интенсивное совершенствование современных информационных технологий и динамичное развитие на их основе глобальной и национальных информационных инфраструктур является одной из характерных примет нашего времени. Во многом благодаря воздействию этого фактора начали складываться условия для постепенного перехода человечества к постиндустриальной фазе своего развития, часто называемой «глобальным информационным обществом». Одним из наиболее важных признаков этого общества, на наш взгляд, является изменение предмета и орудий труда большей части людей. Предметом их труда становятся информация и знания, а орудием труда — информационные технологии. Изменения в общественных отношениях затронут, прежде всего, производственную сферу, которая в значительной степени будет ориентирована на производство продуктов информационной и интеллектуальной деятельности, совершенствование информационных технологий, оказание услуг в создании новой информации и новых знаний. Связанные с этим изменения условий человеческой деятельности неминуемо затронут духовную и социальную сферу жизни общества.

«Глобальное информационное общество» на первом этапе, видимо, будет представлять собой некоторую ассоциацию стран, достигших значительного прогресса в области развития «информационной индустрии», информатизации жизни общества и управления государством, внедрения результатов развития науки, подготовки необходимых кадров, формирования культуры информационной деятельности, интеграции в мировую экономику. Достижение существенного прогресса в этих областях является важным условием экономического процветания государств-членов международного сообщества, и, в конце концов, — сохранения стратегической стабильности в мире.

В то же время переход цивилизации в фазу постиндустриального развития своей обратной стороной имеет объективное усиление зависимости общества от нормального функционирования глобальной и национальных информационных инфраструктур. Очевидно, что «вхождение» тех или иных государств в глобальное информационное общество не отменяет наличия у них национальных интересов, в том числе в информационной сфере, и необходимости обеспечения безопасности этих интересов от угрозы ущемления со стороны других государств, а также таких опасных субъектов современной жизни, как международные террористические организации.

Можно выделить несколько основных источников угроз информационной безопасности общества в современном мире, которые затрагивают как социальные интересы человека, так и интересы общества и государства. На наш взгляд, социальные интересы человека, которые необходимо охранять в информационном обществе, заключаются прежде всего в реальном обеспечении прав и свобод человека на доступ к открытой информации, на использование информации в интересах осуществления не запрещенной законом деятельности, а также в защите информации, обеспечивающей личную безопасность, духовное и интеллектуальное развитие. Наиболее опасным источником угроз этим интересам является существенное расширение возможности манипулирования сознанием человека за счет формирования вокруг него индивидуального «виртуального информационного пространства», а также возможности использования технологий воздействия на его психическую деятельность. Сложность процедур, реализуемых в современных технологиях, критически увеличивает зависимость человека от других людей, осуществляющих разработку информационных технологий, определение алгоритмов поиска требуемой информации, ее предварительной обработки, приведения к виду, удобному

для восприятия, доведение до потребителя. По существу, данные люди во многом формируют для человека информационный фон его жизни. Они определяют «информационные» условия его жизни и деятельности. Именно поэтому представляется исключительно важным обеспечить безопасность взаимодействия человека с информационной инфраструктурой.

Другим опасным источником угроз социальным интересам человека является неправомерное использование персональных данных, накапливаемых различными структурами, в том числе органами государственной власти, а также расширение возможностей скрытого сбора информации, составляющей его личную и семейную тайну, сведений о его частной жизни. Это обусловлено дальнейшими успехами в области миниатюризации средств скрытого сбора и передачи информации. Для противодействия угрозам социальным интересам человека необходимо разработать и реализовать действенные правовые механизмы охраны этих сведений.

Интересы общества в информационной сфере заключаются в обеспечении социальной стабильности и экономического процветания на базе упрочения демократии, поддержания общественного согласия и повышения созидательной активности населения. Одним из источников угроз интересам общества в информационной сфере является непрерывное усложнение информационных и телекоммуникационных систем, сетей связи, информационной составляющей критически важных объектов инфраструктуры жизни общества. Эти угрозы могут проявляться в виде нарушения устойчивости функционирования составляющих информационной инфраструктуры, несанкционированного доступа к охраняемой законом информации экономически и социально значимых структур со стороны преступных, в том числе террористических, организаций. Объектами реализации таких угроз могут выступать информационные системы энергетической, транспортной и некоторых других инфраструктур. Потенциал так называемой «киберпреступности» весьма высок. По имеющимся данным, только за последние три года общее количество зарегистрированных преступлений в сфере компьютерных технологий возросло в России более чем в 150 раз. Тенденция роста этого вида преступлений отмечается и в других странах.

Масштаб возможных последствий нарушения работоспособности технического и программного обеспечения информационных систем можно представить по затратам на решение «Проблемы-2000». По некоторым оценкам, мировое сообщество затратило на эти цели около 500 млрд. долл. США.

Интересы государства в информационной сфере заключаются в использовании информации и информационной инфраструктуры для обеспечения суверенитета и территориальной целостности страны, разъяснения населению страны и международной общественности содержания и направленности государственной политики, в создании условий для гармоничного развития информационной инфраструктуры, в безусловном исполнении законодательства, в поддержании правопорядка, в развитии международного сотрудничества на основе партнерства. Наиболее опасными источниками угроз интересам государства в информационной сфере являются неконтролируемое распространение «информационного оружия» и развертывание гонки вооружений в этой области, попытки реализации концепций ведения «информационных войн». Это обстоятельство особенно опасно в условиях существования почти монопольного положения компаний небольшого количества стран на рынке информационных продуктов, так как способно спровоцировать желание использовать имеющееся превосходство для достижения тех или иных политических целей. Данные угрозы могут проявляться также в виде получения противоправного доступа к сведениям, составляющим государственную тайну, к другой конфиденциальной информации, раскрытие которой может нанести ущерб интересам государства.

В настоящее время завершается подготовка проектов итоговых документов первого этапа Всемирной встречи на высшем уровне по информационному обществу, который состоится в декабре 2003 года в Женеве. Эти документы призваны определить основные принципы и направления сотрудничества в области формирования условий для перехода цивилизации к постиндустриальной фазе развития на значительную перспективу. Именно по этой причине важно, чтобы в них нашли отражение наиболее сложные вопросы этого процесса, к числу которых, конечно, относится создание системы международной информационной безопасности. Основным назначением данной системы должно стать противодействие угрозам использования информационной инфраструктуры в целях, несовместимых с Уставом ООН, и, прежде всего, угрозам:

- манипулирования поведением человека;
- нарушения устойчивости функционирования составляющих информационной инфраструктуры, несанкционированного доступа к охраняемой законом информации со стороны преступных, в том числе террористических, организаций;

- использования «информационного оружия», создаваемого на базе современных информационных технологий, и развертывание гонки вооружений в этой области, реализации концепций ведения «информационных войн».

Актуальность данной проблемы подтверждена Генеральной Ассамблеей ООН, которая, начиная с 1998 года, по инициативе Российской Федерации ежегодно консенсусом принимает резолюции «Достижения в сфере информатизации и телекоммуникаций в контексте международной безопасности». Данные резолюции с каждым годом становятся все более «мускулистыми». Мировое сообщество тем самым признает обеспечение международной информационной безопасности, как глобальную проблему, решение которой представляет одну из важных сфер международного сотрудничества.

Бурное развитие информационной сферы неизбежно оказывает влияние и на формирование внутренней политики государства, что требует постоянной адаптации существующих государственных и общественных институтов к этим инновациям. Содержание государственной информационной политики требует своей глубокой и комплексной проработки. Под воздействием общемирового интенсивного развития информационных технологий заинтересованные органы государственной власти страны с участием ученых и специалистов призваны активно разрабатывать и внедрять концептуальные и программные основы, принципы, а также практические приложения в этой области.

Успешность противодействия угрозам в информационной сфере во многом зависит от того, насколько эффективно будет использован потенциал, который накоплен российской наукой в области решения научно-технических проблем информационной безопасности и, прежде всего, математических проблем обеспечения безопасности информационных технологий. Полагаю, что свою лепту в становление России как информационной державы внесет недавно созданный в системе Московского государственного университета Институт проблем информационной безопасности.

Математические проблемы безопасности информационных технологий

В. А. Садовничий, В. А. Носов, В. В. Ященко

В декабре нынешнего года в Женеве состоится очередной саммит «большой восьмерки». Он будет посвящен проблемам построения глобального информационного общества. Новые вызовы и угрозы, с которыми сталкивается человечество на пути построения такого общества, раскрыты в докладе В. П. Шерстюка. Современные компьютеры, глобальные информационные сети и сетевые технологии сильно изменили нашу жизнь, но вместе с новыми возможностями у нас появились и новые риски. «Как пользоваться такими возможностями, но при этом нейтрализовать риски или хотя бы снизить возможный ущерб от их реализации?» — вот главный вопрос обеспечения безопасности информационных технологий. Для ответа на этот вопрос необходимо решить большое количество разнообразных задач: и политических, и экономических, и научных, и технических.

Официальная позиция государства по направлениям решения указанных задач изложена в Доктрине информационной безопасности Российской Федерации, утвержденной в сентябре 2001 года. Секция Научного Совета при Совете Безопасности Российской Федерации на основе Доктрины разработала «Перечень приоритетных научных проблем в области информационной безопасности». Многие из проблем, включенных в Перечень, являются по сути своей междисциплинарными, и для их решения потребуется объединение усилий математиков, компьютерщиков, физиков, юристов, социологов, экономистов. Некоторые проблемы — чисто математические или чисто технические. Все участники конференции получили Перечень проблем вместе с программой конференции и поэтому имеют возможность внимательно изучить его. Приведу несколько проблем, наиболее близких участникам конференции:

54. Разработка фундаментальных проблем теоретической криптографии и смежных с ней областей математики.

56. Разработка и обоснование новых методов криптографического анализа современных шифрсистем.

57. Разработка перспективных криптографических протоколов взаимодействия абонентов в сложных иерархических глобальных сетях и распределенных информационно-аналитических системах.

65. Разработка проблем создания технических средств обработки информации, защищенных от физико-технических методов информационного доступа.

74. Разработка, теоретическое и экспериментальное исследование современных методов стеганографии, других средств тайнописи и защиты от подделки.

В нашем докладе будут затронуты не только конкретные математические проблемы информационной безопасности, но и некоторые организационно-научные вопросы, связанные с этими проблемами.

Наиболее действенным инструментом обеспечения безопасности информационных технологий является криптография. Год назад на конференции «Московский университет и развитие криптографии в России» состоялся подробный разговор о развитии криптографии, как науки, в XX веке и о влиянии криптографии на развитие математики. Хотелось бы продолжить этот разговор и подчеркнуть несколько особенностей развития криптографии в последние годы.

Во-первых, под влиянием процессов глобальной информатизации резко расширился круг заказчиков и потребителей криптографической продукции. Поэтому в криптографии происходит «размежевание» идей, методов и результатов, предназначенных для защиты государственных секретов, и, так сказать, «для массового потребления». Конечно же, такое размежевание необходимо и естественно,

поскольку у заказчиков принципиально разные требования как по уровню защиты информации, так и по стоимости защиты.

Во-вторых, постоянно растет количество публикаций по криптографии (и научных, и псевдонаучных). Это происходит не только из-за расширения круга заказчиков, но также и вследствие стремительного роста числа приложений криптографии и новых научных задач, инициированных криптографией. Поскольку раньше базовые криптографические знания были секретными, то даже вполне квалифицированные ученые, подключаясь к решению таких новых задач, допускают вполне объяснимые криптографические ошибки. Следовательно, нужна открытая (в допустимых пределах) криптографическая литература — и фундаментальные монографии, и учебники, и научно-популярная литература. Следует отметить, что за последние 5 лет в стране уже вышло несколько серьезных книг по криптографии, но этого явно недостаточно.

В-третьих, активно развивается международное научное сотрудничество по проблемам информационной безопасности. Так, Московский университет уже 3 года сотрудничает с Консорциумом военных академий и институтов, изучающих проблемы безопасности. В 2001 году в МГУ прошла конференция по информационной безопасности, в которой приняло участие более 500 ученых и специалистов из 45 стран. После конференции наши ученые участвовали в нескольких международных мероприятиях по проблемам информационной безопасности и борьбы с кибертерроризмом. Наш опыт международного научного сотрудничества в этой области показал, что иногда мы с зарубежными коллегами говорим на разных языках. Это отчасти объясняется тем, что до сих пор не выработана единая международно-признанная терминология. Например, существуют разные трактовки содержания и соотношения понятий «информационная безопасность», «безопасность информационных технологий», «безопасность информации», «безопасность киберпространства». Московский университет выступил с инициативой разработки многоязычного глоссария по проблемам безопасности информационного общества, и в настоящее время ведутся переговоры о создании международного коллектива для проведения этой работы. Одновременно с этим мы совместно с Академией криптографии Российской Федерации разрабатываем открытый русско-английский и англо-русский словарь криптографических терминов. Это тяжелая и очень объемная работа, в ней много дискуссионных вопросов, уходящих корнями и в историю, и в секретность. Тем не менее, мы надеемся в следующем году издать такой словарь.

Несколько слов о современном состоянии открытых математических направлений криптографии. Год назад на нашей конференции было представлено несколько обзорных докладов по таким направлениям, сегодня мы продолжим эту практику. Хотелось бы подчеркнуть одну важную особенность последних лет: наряду с развитием традиционных математических направлений криптографии (алгебраическое, теоретико-числовое, комбинаторное и т. д.) происходит формирование нового направления, которое можно назвать математической криптографией. Базовыми понятиями математической криптографии являются односторонняя функция, псевдослучайный генератор и доказательство с нулевым разглашением. Они позволяют строить новые математические модели, которые более адекватно отражают задачи криптографии. При этом возникают чисто математические постановки задач, а математическая криптография получает свой собственный аппарат и свои внутренние стимулы для развития. Естественно, что основные задачи математической криптографии изначально сложны, а некоторые из них по сути своей являются давно известными нерешенными математическими проблемами. Например, старая проблема нижних оценок вычислительной сложности — серьезные продвижения в ее решении позволят обосновать стойкость математических моделей большинства шифров. Вообще, математическая криптография — это наука, главным образом, о вычислительной сложности задач из некоторых специальных классов. Другой пример. Уже четверть века известно несколько гипотетических функций с секретом, основанных на задаче факторизации целых чисел. И до сих пор остается открытым вопрос о существовании еще хотя бы одной такой функции, основанной на какой-либо иной математической задаче.

Много важных теоретических и прикладных задач математической криптографии связано с изучением и обоснованием стойкости криптографических протоколов. В настоящее время выделено уже несколько десятков криптографических протоколов. Среди них есть и простые криптографические протоколы, например, протокол подбрасывания монеты по телефону. Есть и намного более сложные:

- протокол распределенного конфиденциального вычисления,
- протокол голосования,

- система электронных платежей.

В этой бурно развивающейся и важной для приложений области математической криптографии еще многое не ясно. Отметим, например, что для некоторых протоколов даже не сформулированы модели атак и угроз.

В последнее время большие надежды возлагаются на криптографию, использующую квантовый канал связи, или, как ее принято называть, квантовую криптографию. О квантовой криптографии пишут и в газетах, и в научно-популярной литературе. Международные научные коллективы проводят исследования и эксперименты по реализации квантового канала связи и периодически сообщают о своих очередных достижениях. Математики разрабатывают и обосновывают стойкость различных криптографических протоколов, главным образом, протоколов генерации ключей с использованием гипотетического квантового канала связи. Это направление действительно очень перспективное, и для его реализации необходимы согласованные усилия физиков, математиков, криптографов.

Относительно новым направлением обеспечения безопасности информационных технологий является компьютерная стеганография. Разработанные в стеганографии методы скрытого «встраивания» одной информации в другую позволяют решать много новых задач. Так, например, имеется много публикаций по разработке идей построения электронных «водяных знаков» и электронных «отпечатков пальцев» методами стеганографии с использованием методов криптографии. Пока они носят чисто математический характер. Но если эти математические идеи удастся довести до технологии, то будут решены важные задачи защиты авторских прав и борьбы с незаконным распространением копий. Такие результаты очень нужны разработчикам программных продуктов.

В заключение хочется отметить, что в программе конференции — доклады по различным математическим направлениям. Надеемся, что состоится полезный обмен мнениями и исследования математических проблем безопасности информационных технологий получат новый импульс.

Приоритетные проблемы научных исследований в области информационной безопасности Российской Федерации[1]

Одобрены секцией по информационной безопасности научного совета при Совете Безопасности Российской Федерации (протокол от 28 марта 2001 г. № 1).

Гуманитарные проблемы обеспечения информационной безопасности Российской Федерации

1. Исследование места и роли проблем информационной безопасности в становлении современного информационного общества.

2. Исследование проблем обеспечения баланса интересов личности, общества и государства в информационной сфере.

3. Исследование роли и места информационной безопасности в обеспечении военной, экономической, экологической, иных видов национальной безопасности.

4. Разработка единого понятийного аппарата (терминов и определений) в сфере информационной безопасности.

5. Научное обоснование основных направлений деятельности государственных ведомственных структур по обеспечению информационной безопасности Российской Федерации.

6. Национальные интересы России и информационное противостояние в современном мире.

7. Ценностная ориентация личности, ее информационное обоснование и информационная безопасность.

8. Информационная безопасность и политическая этика.

9. Информационное пространство и проблема целостности российского государства.

10. Изучение и прогнозирование социально-психологических последствий внедрения и широкого распространения современных информационных технологий.

11. Исследование исторических аспектов, современного состояния и возможности развития информационной деятельности зарубежных государств с использованием ими российских информационных систем для пропаганды своих интересов.

12. Разработка и научное обоснование системы мониторинга состояния информационной безопасности Российской Федерации.

[1] Приложение к докладу В. А. Садовничего, В. А. Носова, В. В. Ященко «Математические проблемы безопасности информационных технологий».

13. Разработка информационно-динамической модели баланса между потребностью в свободном обмене информацией и допустимыми ограничениями ее распространения.

14. Разработка правовых механизмов обеспечения конституционных прав и свобод граждан в информационной сфере.

15. Проблемы правовой охраны, распределения прав собственности и прибыли (доходов) по результатам научно-технической деятельности, по вознаграждению авторов и лиц, содействующих использованию объектов интеллектуальной собственности.

16. Исследование места и роли СМИ в решении задач информационного обеспечения государственной политики Российской Федерации.

17. Развитие нормативной базы, направленное на сохранение и правовую защиту российской интеллектуальной собственности в информационной сфере.

18. Совершенствование правового обеспечения, регламентирующего создание и использование банков данных, а также иных информационных ресурсов, имеющих федеральное значение.

19. Разработка проблем правового регулирования в области технологической независимости.

20. Разработка механизма правового регулирования защиты и использования технологий двойного применения.

21. Разработка моделей и механизмов страхования информационных рисков.

22. Разработка правовых механизмов сотрудничества государств-участников СНГ в обеспечении коллективной информационной безопасности.

23. Разработка проблем правового регулирования в вопросах инвестиционной политики в области информационных технологий.

24. Разработка правовых механизмов регулирования в сфере производства и эксплуатации криптографических продуктов.

25. Разработка правовых механизмов регулирования электронного документооборота.

26. Проблемы правового обеспечения создания и функционирования системы мониторинга угроз информационных атак на критически важные сегменты информационной инфраструктуры Российской Федерации.

27. Совершенствование нормативно-методической базы проведения экспертизы и контроля качества защиты информации и информационных ресурсов.

28. Разработка международно-правовых механизмов сдерживания информационного противоборства.

29. Гармонизация отечественных и зарубежных стандартов в области информационных технологий.

30. Проблемы формирования международной системы информационной безопасности.

31. Разработка моделей и правовых механизмов взаимодействия Центра и субъектов Российской Федерации в информационной сфере.

32. Разработка моделей и правовых механизмов взаимодействия органов власти субъектов Российской Федерации и органов местного самоуправления в информационной сфере.

33. Разработка и научное обоснование путей обеспечения информационно-психологической безопасности личности и общества.

Научно-технические проблемы обеспечения информационной безопасности Российской Федерации (физико-математические, технические)

34. Разработка концептуальной взаимоувязанной структуры информационного пространства и состава информационных ресурсов.

35. Проблемы создания и развития информационной составляющей информационно-телекоммуникационной системы специального назначения в интересах органов государственной власти (ИТКС).

36. Исследование проблем обеспечения информационной безопасности национальных платежных систем на базе российских интеллектуальных карт.

37. Исследование проблем создания и развития национальной системы управления цифровыми сертификатами.

38. Поиск путей решения проблемы создания единой системы технических стандартов информационного обмена (протоколов, форматов данных, спецификаций интерфейсов) с учетом существующих международных стандартов и перспектив их развития.

39. Исследование подходов к созданию отечественной системы промышленных стандартов проектирования и разработки информационных систем и систем телекоммуникаций с учетом существующих международных стандартов и перспектив их развития.

40. Исследования, направленные на создание комплекса отечественных инструментальных средств проектирования информационных систем.

41. Проблемы совершенствования отечественного программного обеспечения.

42. Разработка и обоснование систем сертификации средств, содержащих элементы импортного производства.

43. Анализ возможности использования технологических особенностей производства новейших зарубежных и отечественных образцов элементной базы микроэлектроники для реализации деструктивных информационных функций.

44. Исследование проблем создания и функционирования национального эталонного банка доверенного программного обеспечения.

45. Исследование проблем создания и развития защищенных информационно-телекоммуникационных систем, в том числе разработка методов выбора архитектуры и расчета параметров этих систем, математических моделей и технологий управления, системного и прикладного программного обеспечения с интеграцией функций защиты, средств взаимодействия, устройств передачи и распределения информации.

46. Разработка моделей угроз безопасности систем и способов их реализации, определение критериев уязвимости и устойчивости систем к деструктивным воздействиям, разработка методов и средств мониторинга для выявления фактов применения несанкционированных информационных воздействий, разработка методологии и методического аппарата оценки ущерба от воздействия угроз информационной безопасности.

47. Разработка методов и средств проведения экспертизы и контроля качества защиты информации и информационных ресурсов, в том числе вопросов оценки базовых общесистемных программных средств на соответствие требованиям информационной безопасности.

48. Разработка методов и средств обеспечения информационной безопасности информационных и телекоммуникационных систем, в том числе автоматизированных систем управления безопасностью,

методов и средств распределения ключей и защиты информации и информационных ресурсов от несанкционированного доступа и разрушающего информационного воздействия, антивирусных технологий, методов и средств контроля состояния защищенности от НСД современных и перспективных технических средств и каналов связи, решение проблемы гарантированного уничтожения остаточной информации на магнитных носителях, исследование и развитие методов построения защищенных систем, использующих ненадежные (с точки зрения информационной безопасности) элементы, включая проблему их тестирования.

49. Исследование проблем безопасности общероссийской информационной инфраструктуры в условиях ее вхождения в глобальные инфраструктуры.

50. Исследование проблем обеспечения информационной безопасности ИТКС, в том числе разработка нормативно-технической документации по безопасности, автоматизированных систем управления безопасностью, унифицированного ряда средств криптографической защиты с учетом используемых в ИТКС технологий обработки информации.

51. Исследование проблем информационной безопасности корпоративных сетей, в том числе сетей науки и образования (в рамках комплексной программы Минпромнауки России «Научное, научно-методическое, материально-техническое и информационное обеспечение системы образования»).

52. Проблемы лицензирования деятельности в области информационно-телекоммуникационных систем.

53. Анализ тенденций в развитии глобальной информационной сети и состояния участия в ней России.

54. Разработка фундаментальных проблем теоретической криптографии и смежных с ней областей математики.

55. Разработка криптографических проблем создания перспективных отечественных шифрсистем (в частности, высокоскоростных).

56. Разработка и обоснование новых методов криптографического анализа современных шифрсистем.

57. Разработка перспективных криптографических протоколов взаимодействия абонентов в сложных иерархических глобальных сетях и распределенных информационно-аналитических системах.

58. Исследование существующих и разработка новых систем с открытым ключом, соответствующих этим системам схем аутентификации и электронной цифровой подписи.

59. Совершенствование нормативно-методической базы по вопросам защиты информации с применением криптографических средств.

60. Анализ основных направлений и тенденций развития отечественных и зарубежных средств криптографической защиты информации.

61. Анализ возможности использования достижений физики и техники для получения доступа к информации, обрабатываемой на современных технических средствах, в том числе исследование физических основ утечки информации от технических средств по побочным каналам, разработка проблем аналитической обработки побочных сигналов.

62. Исследование алгоритмических и технологических особенностей новейших зарубежных и отечественных технических средств обработки информации.

63. Исследование проблем и методов информационного доступа к каналам связи.

64. Разработка методологии оценивания защищенности, комплексных методов и средств защиты технических средств обработки информации от физико-технических методов несанкционированного доступа, совершенствование соответствующей нормативной базы.

65. Разработка проблем создания технических средств обработки информации, защищенных от физико-технических методов информационного доступа.

66. Сравнительный анализ тенденций развития физико-технических проблем защиты информации в стране и за рубежом.

67. Исследование архитектурных вариантов построения вычислительных систем высокой производительности, алгоритмического и программного обеспечения с учетом особенностей криптографических задач.

68. Исследование проблем построения автоматизированных систем обработки криптографической информации в неоднородной вычислительной среде.

69. Исследование проблем управления распределенными вычислительными процессами.

70. Разработка и научное обоснование моделей угроз и стратегий защиты объектов от технических разведок.

71. Разработка методов и средств противодействия техническим разведкам с учетом эффективности их функционирования.

72. Разработка методов и средств контроля состояния и достаточности принимаемых мер по противодействию техническим разведкам на объектах защиты.

73. Разработка современной методологии обеспечения противодействия техническим разведкам на объектах защиты.

74. Разработка, теоретическое и экспериментальное исследование современных методов стеганографии, других средств тайнописи и защиты от подделки.

75. Исследование и разработка отечественных защитных экранов с учетом моделей угроз для уже существующих и перспективных цифровых АТС.

Проблемы кадрового обеспечения информационной безопасности Российской Федерации

76. Обоснование облика, структуры и путей реализации единой системы подготовки кадров в области современных информационных технологий и информационной безопасности.

77. Обоснование структуры и функций Учебно-методического комплекса по подготовке, повышению квалификации и переподготовке кадров в области информационной безопасности.

78. Разработка государственных образовательных стандартов по новым специальностям высшего профессионального образования.

79. Создание нормативно-правовой базы особого порядка лицензирования образовательной деятельности в области информационной безопасности.

80. Проблемы нормативно-правового обеспечения подготовки специалистов по вопросам информационной безопасности и смежных областях.

81. Развитие нормативной базы, направленной на сохранение интеллектуального потенциала государственных вузов Российской Федерации, осуществляющих подготовку специалистов в области современных информационных технологий и информационной безопасности.

82. Разработка методик, специальной и учебной литературы по специальностям в области информационной безопасности, включая разработку учебных пособий для подготовки специалистов в области криптографии.

83. Разработка методик, специальной и учебной литературы по изучению общих вопросов информационной безопасности в специальностях, не отнесенных к группе «Информационная безопасность».

84. Разработка базового мультимедийного учебно-методического комплекса по подготовке специалистов в области информационной безопасности и информационного противоборства.

85. Разработка методик, специальной и учебной литературы для курсов переподготовки и повышения квалификации кадров в области информационной безопасности.

86. Программные и аппаратные средства реализации современных информационных технологий в образовательном процессе.

87. Проблема использования в образовательном процессе деловых и специальных исследовательских игр по информационной безопасности.

Проблемы безопасности программного обеспечения

В. Б. Бетелин

1 Вторичность требований безопасности

В современной информационной системе (ИС) можно выделить две основные компоненты: *прикладную* и *инфраструктурную*.

Первая реализует функциональность, присущую только данной ИС (или указанному классу ИС), предоставляя соответствующие прикладные сервисы. Вторая является фундаментом первой, предоставляя для нее системные сервисы и обеспечивая наличие определенных свойств, необходимых для успешной разработки, эксплуатации и модернизации ИС. Важнейшим из таких свойств является информационная безопасность (ИБ), уровень которой для данной ИС зависит, в первую очередь, от ее инфраструктурной компоненты. Эта компонента включает следующие основные уровни:

- оборудование для передачи данных (кабельные системы, радиосети и т. д.);
- локальные сети;
- глобальные сети;
- аппаратная (компьютерная) платформа;
- системные сервисы (программная платформа).

С точки зрения информационной безопасности самым слабым звеном инфраструктурной компоненты является аппаратная (компьютерная) и программная платформы, которые, как правило, создаются на базе массовых коммерческих продуктов (технологические пакеты, СУБД, операционные системы, персональные ЭВМ и т. д.). Как и для любых других производителей товаров массового спроса, конкурентная борьба на этих рынках требует от производителя реализации стратегии «двойного сокращения»:

- сокращение времени жизни производимого продукта;
- сокращение сроков разработки нового продукта с новыми функциональностями.

Существенно, что *первичность требований рынка, очевидно, вступает в противоречие с требованиями обеспечения информационной безопасности ИС*, поскольку:

- повышение быстродействия микропроцессоров и СБИС, развитие архитектур с высокой степенью параллелизма позволяет методом грубой силы преодолевать барьеры (прежде всего криптографические), ранее казавшиеся неприступными;
- развитие сетей и сетевых технологий, увеличение числа связей между информационными системами, рост пропускной способности каналов расширяют круг злоумышленников, имеющих техническую возможность организовать атаки;
- появление новых информационных сервисов ведет к появлению новых уязвимостей как «внутри» сервисов, так и на их стыках;
- конкуренция среди производителей элементной базы, ЭВМ и программного обеспечения заставляет сокращать сроки разработки, что ведет к снижению качества тестирования и выпуску продуктов с дефектами защиты;

- навязываемая потребителям парадигма постоянного наращивания возможностей аппаратного и программного обеспечения не позволяет долго оставаться в рамках надежных, апробированных конфигураций и, кроме того, в силу бюджетных ограничений, заставляет снижать долю ассигнований на безопасность.

2 Возрастающая сложность программной платформы

Одной из количественных характеристик сложности программной системы является минимальный объем ОЗУ, необходимый для ее функционирования, и минимальный размер области на внешнем накопителе, необходимый для размещения дистрибутива этой системы. Для операционной системы MSDOS, которую фирма Microsoft развивала и поддерживала в течении десяти лет (1984–1994), требовалось не менее 640 Кбайт ОЗУ и не менее 10 Мбайт на внешнем накопителе. Для операционной системы Windows 95, анонсированной в 1995 году, требовалось уже не менее 8 MB ОЗУ и не менее 48 Мбайт на внешнем накопителе, то есть более чем в 10 и 4.8 раза больше соответственно.

Операционная система Windows 2000 требует уже 64–256 Мбайт ОЗУ и 0.65–1 Гбайт на внешнем накопителе, то есть минимум в 8 и 10 раз соответственно больше чем для Windows 95. Приведенные выше требования заявлены производителем, эти требования заметно занижены, фактические требования заметно выше.

Дэвид Корн, автор Korn-shell — одного из наиболее известных командных интерпретаторов системы Unix, в течение двух лет занимался реализацией интерфейса прикладного программиста (API) операционной системы UNIX на основе API Win32. В статье «Working with Win32: the Good, the Bad, and the Ugly» [3] (русский перевод выполнен С. Кузнецовым) среди плохих аспектов Win32 Корн называет прежде всего

а) сложность интерфейса, и

б) сложную модель защиты,

которые, по его мнению, и являются основной причиной проблем с безопасностью. Параллельно в статье отмечается и количественная сложность интерфейса. Действительно, библиотека ядра включает 675 функций; библиотека, определяющая интерфейс безопасности, содержит около 400 функций; библиотека, обеспечивающая пользователя, содержит примерно 600 функций.

В 1993 году объем исходного кода ядра ОС Linux 1.0 составлял 180 тыс. строк, а для его размещения на внешнем носителе требовалось 6 Мбайт. В 2003 году объем исходного кода ядра ОС Linux 2.6.0 составляет уже 6 млн. строк (в 30 раз больше), и для его размещения на внешнем носителе требуется уже 212 Мбайт (в 35 раз больше).

Аналогичным образом усложняются и инструментальные пакеты. Например, для пакета Microsoft Office 95 минимально необходимый объем на внешнем носителе составлял 55 Мбайт, для Office 2003 требуется уже 500 Мбайт, т. е. на порядок больше.

3 Сложность программных платформ — основное препятствие на пути обеспечения их безопасности

3.1 Краткий обзор проблем безопасности систем на базе платформ Linux и Windows

Об этом прежде всего свидетельствуют данные фирм Microsoft и Red Hat о количестве выпущенных в 1998–2003 гг. извещения об ошибках (табл. 1).

Проблемы безопасности программных платформ волнуют как государственные структуры развитых стран, так и мировое сообщество специалистов по информационным технологиям. В 2001 году ФБР был опубликован список 20 сервисов, наиболее критичных с точки зрения безопасности интернет-систем. Этот список содержал по 10 наиболее часто атакуемых сервисов и приложений для платформ Windows и Unix. По данным CERN в 2001 году было обнаружено более 3 тысяч уязвимостей в различных программных продуктах.

Red Hat Linux

Дистрибутив	Год выпуска	Количество извещений об ошибках
8.0	2002	113 извещений
7.3	2001	158 извещений
6.2	2000	110 извещений

Microsoft Security Bulletins

Год	Количество извещений
2002	72
2001	60
2000	100
1999	60
1998	20

Таблица 1:

Из материалов компаний MI2G и SIP&S следует, что в период январь–ноябрь 2002 г. было зарегистрировано 1162 уязвимости в операционных системах, серверном ПО и приложениях. Из этого числа более 500 относится к платформе Intel-Windows и более 200 к ОС Linux.

В 2002 году была обнаружена серьезная уязвимость, проявляющаяся одновременно в нескольких ОС фирмы Microsoft: Windows NT 4.0, Windows 98, Windows 98SE, Windows Me и Windows 2000. Эта уязвимость дает злоумышленнику возможность несанкционированного форматирования дисков на стороне сервера (22.11.2002, Microsoft, извещение N65). В 2003 году были обнаружены еще более серьезные уязвимости в Linux, дающие возможность динамической подмены ядра извне (Silvio Cesare kernel infection).

В мае 2003 года была обнаружена критическая уязвимость в системе Microsoft.Net Passport, которой регулярно пользуются около 200 млн. человек для «надежного» хранения конфиденциальной информации.

Наконец, в августе 2003 года свыше 330 тыс. Windows XP и Windows 2000 установок оказались пораженными программой MSBlast.

3.2 Анализ информационной безопасности систем на платформах Linux и Solaris

Профессиональные платформы имеют те же проблемы с обеспечением безопасности, что и рассмотренные выше платформы. Ниже приведены результаты сравнительного анализа защищенности свободно распространяемой ОС Linux и профессиональной ОС Solaris.

Объекты анализа и методика

Анализ защищенности был проведен для следующих версий операционных систем:

- ОС Linux Red Hat версии 7.3, версия ядра 2.4.18-3;
- ОС Solaris 2.5.

Для анализа защищенности использовался свободно распространяемый продукт *Nessus* версии 2.0.7 (см. `http://nessus.org`). Он характеризуется обширностью (более полутора тысяч элементов) и актуальностью базы данных осуществляемых проверок.

Nessus выявляет уязвимости как путем пассивного анализа, изучая конфигурационные файлы, задействованные порты и т. п., так и путем активного опробования, имитируя действия, выполняемые атакующими. Основными группами проверок являются:

- возможность несанкционированного получения привилегий суперпользователя;
- возможность несанкционированного выполнения команд;

- возможность несанкционированного доступа к файлам.

При проведении исследования были активированы все методы проверки, в том числе и те, которые могут повлиять на работу проверяемых систем.

Результаты анализа

Далее представлены основные результаты проведенного анализа для каждой из перечисленных выше операционных систем.

На *ОС Linux* обнаружено более полутора десятков открытых портов, на части которых предоставляются небезопасные сервисы.

Выявлены следующие уязвимости с высоким и средним уровнем риска:

- группа уязвимостей, ассоциированных с портом 22/tcp: использование версии OpenSSH, старше чем 3.2.1, позволяет удаленно получить несанкционированный доступ к командному интерпретатору и, при локальном доступе, получить права суперпользователя (для ранних версий права суперпользователя можно получить и при удаленном доступе);
- наличие сервиса rexecd, не имеющего надежных средств аутентификации и позволяющего злоумышленнику сканировать порты других компьютеров;
- наличие уязвимости по отношению к переполнению буфера в X-сервисе шрифтов (порт 7100/tcp), позволяющей получить права суперпользователя при удаленном доступе;
- наличие файловых систем, общедоступных для монтирования по nfs, что позволяет осуществить несанкционированный доступ к файлам;
- наличие уязвимости в используемой версии nfsd, позволяющей выполнять произвольные команды;
- наличие уязвимости по отношению к атакам с помощью цепочек формата в используемой версии сервиса statd, которая дает возможность злоумышленнику выполнять произвольные команды;
- использование небезопасного протокола XDMCP, позволяющего перехватывать все нажатия на клавиатуре X-терминала (включая ввод паролей);
- наличие уязвимости в реализации протокола ICMP, открывающей доступ к фрагментам памяти ядра.

В системе на базе ОС Solaris 2.5 обнаружено более 50 открытых портов, на части которых предоставляются следующие небезопасные сервисы:

- многочисленные серьезные уязвимости выявлены в сервисе smtp (25/tcp), связанные, в частности, с переполнением буфера, что может быть использовано для получения привилегий суперпользователя и выполнения небезопасных команд;
- SMTP-сервер может быть использован для перенаправления почты, что позволяет организовать с его помощью массовые «спам»-рассылки;
- путем передачи ненормально большого количества параметров (переменных окружения) в программу /bin/login можно заставить ее аварийно завершиться;
- использование версии OpenSSH, старше чем 3.4, имеет известные уязвимости;
- наличие уязвимости FTP-сервера, позволяющей локальным пользователям получить доступ к теневому файлу паролей;
- небезопасная конфигурация сервера имен;
- возможность воспользоваться пустыми входными параметрами для netbios-ssn (139/tcp);
- уязвимость Samba-сервера по отношению к переполнению буфера;

- наличие сервера rexecd;
- наличие файловых систем, общедоступных для монтирования по nfs;
- наличие уязвимости в используемой версии nfsd, позволяющей выполнять произвольные команды;
- наличие уязвимости по отношению к атакам с помощью цепочек формата в используемой версии сервиса statd;
- наличие уязвимости по отношению к переполнению буфера в X-сервисе шрифтов (7100/tcp);
- наличие сервиса Kodak Color Management System (32774/tcp), который на Solaris 2.5 позволяет пользователям записывать произвольные файлы и получать привилегии суперпользователя;
- наличие сервиса tooltalk RPC (32773/tcp), уязвимого по отношению к атакам с помощью цепочек формата, позволяющим удаленным пользователям получать привилегии суперпользователя;
- допустимость пакетов TCP SYN с установленным флагом FIN, что позволяет обходить правила фильтрации межсетевых экранов;
- наличие уязвимости в реализации сервиса sadmin RPC на ОС Solaris, позволяющей выполнять произвольные команды;
- наличие уязвимости в реализации сервиса rpc.walld RPC на ОС Solaris версий 2.5.1, 2.6, 7 и 8 позволяющей получать привилегии суперпользователя;
- наличие сервиса bootparamd RPC, позволяющего получить доступ к файлу паролей NIS.

Устранение указанных уязвимостей тщательным конфигурированием уже установленных экземпляров ОС Linux и ОС Solaris представляется весьма проблематичным ввиду большого числа используемых инсталяций, разнородности требований к ним и постоянным обнаружением новых проблем как в старых, так и в новых версиях ОС.

3.3 Возрастание сложности аппаратно-программной платформы — прямое следствие прогресса в области проектирования и производства СБИС

Одним из наиболее важных технологических параметров, определяющих положение фирмы на мировом микроэлектронном рынке являются значения минимальных норм проектирования производимых этой фирмой изделий. Меньшие, чем у конкурентов, проектные нормы обеспечивают возможность размещения на единице площади кремния большее число транзисторов. Для чипов оперативной памяти меньшие проектные нормы означают уменьшение стоимости хранения одного бита информации, а для микропроцессоров — реализацию дополнительных функциональных возможностей. Кроме того, меньшие проектные нормы позволяют обеспечить функционирование изделий на более высокой рабочей частоте, то есть в общем случае повысить скорость обработки данных. Так, например, с проектными нормами 0.5 мкм. в 1991 году производились 4 Мбит чипы динамической памяти с площадью кристалла 44 кв. мм. Через одиннадцать лет в 2002 году с проектными нормами 0.13 мкм. производились 256 Мбит чипы динамической памяти с площадью кристалла 88 кв. мм. Таким образом, за последние десять лет емкость на единицу площади увеличилась примерно в 32 раза. Аналогично за последние десять лет примерно на порядок увеличились как сложность микропроцессоров (от 4–5 до 40–50 млн. транзисторов), так и их рабочая частота (от 200–300 МГц до 2–3 ГГц). Соответственно увеличились функциональные возможности микропроцессоров.

Прямым следствием этого постоянного роста является увеличение совокупной сложности аппаратно-программной платформы, которое собственно и оказывается основным препятствием на пути обеспечения безопасности.

Ожидается, что к 2010 году сложность микропроцессоров возрастет до 1–1.5 млрд. транзисторов, их рабочая частота увеличится до 30 ГГц, а емкость одного чипа памяти достигнет 8 Гбит. По оценкам экспертов следствием этого будет как минимум десятикратное увеличение объема ядра ОС Linux — 30–40 млн. строк исходного кода. Конечно, неизбежно значительное увеличение сложности и других коммерческих программных и аппаратных платформ и, *следовательно, сложность как объективное следствие конкурентной борьбы на микроэлектронном рынке по-прежнему останется основной проблемой на пути обеспечения безопасности этих платформ.*

4 Безопасная аппаратно-программная платформа

4.1 Основные положения

Решение основной проблемы возможно путем *создания аппаратно-программной платформы, для которой требования обеспечения информационной безопасности являются первичными, основополагающими [1, 2].*

Первичность требования обеспечения информационной безопасности аппаратно-программной платформы в данном случае означает:

- самоограничение минимумом функциональности (и аппаратуры, и операционной системы), достаточным для реализации и функционирования сервисов безопасности;
- следование стандартам, использование апробированных решений (повышает надежность системы, уменьшает возможность попадания процесса разработки платформы в тупиковую ситуацию, когда обеспечение безопасности потребует непомерно больших затрат и принципиальных модернизаций);
- простота реализации (только для простого средства можно формально или неформально доказать его корректность).

Сервисы безопасности, какими бы мощными и стойкими они ни были, сами по себе не могут гарантировать надежность программно-технического уровня защиты. Только разумная, проверенная архитектура способна сделать эффективным объединение сервисов, обеспечить управляемость информационной системы, ее способность развиваться и противостоять новым угрозам при сохранении таких свойств, как высокая производительность, простота и удобство использования.

Такая архитектура должна основываться на следующих трех принципах:

- необходимость выработки и проведения в жизнь единой политики безопасности;
- необходимость обеспечения конфиденциальности и целостности при сетевых взаимодействиях;
- необходимость формирования составных сервисов по содержательному принципу, чтобы каждый полученный таким образом компонент обладал полным набором защитных средств и с внешней точки зрения представлял собой единое целое (не должно быть информационных потоков, идущих к незащищенным сервисам).

4.2 Краткая характеристика ос2000

Существенны следующие позитивные (описывающие то, что присутствует в ос2000) свойства:

- следование международному стандарту POSIX 1003.1 в части, касающейся программного интерфейса к средствам ввода/вывода, механизмам реального времени, средствам протоколирования;
- следование международному стандарту языка Си в плане поддержки среды времени выполнения;
- следование спецификациям семейства протоколов TCP/IP, наличие реализации протоколов прикладного уровня: FTP (клиент), TELNET (сервер и клиент), NFS (сервер и клиент);
- развитые средства конфигурирования, возможность отбора только тех элементов ОС, которые необходимы для работы приложений.

Не менее существенны негативные свойства (описывающие то, что в ос2000 отсутствует):

- отсутствие поддержки двух режимов работы микропроцессоров, использование только привилегированного режима;
- отсутствие поддержки механизма виртуальной памяти и деления адресного пространства на системное и пользовательское;
- отсутствие понятия процесса, функционирование всех потоков в одном адресном пространстве;

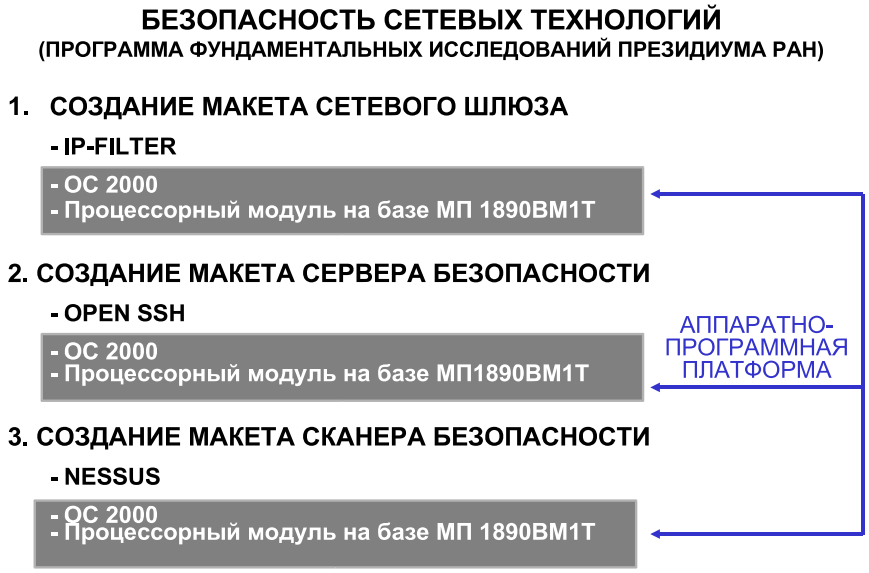


Рис. 1:

- отсутствие понятия пользователя и, как следствие, отсутствие средств разграничения доступа.

Основное следствие позитивных свойств — мобильность программного обеспечения (ПО), следующего стандартам , в сочетании с достаточно широким спектром функциональных возможностей.

Следствие негативных свойств — простота и компактность ОС, минимизация числа возможных ошибок и уязвимостей. Отметим в этой связи аккуратность реализации стека протоколов TCP/IP. Система анализа защищенности Nessus обнаружила лишь две потенциальные уязвимости с низким уровнем риска: поддержка небезопасного протокола TELNET, предсказуемость идентификаторов IP-пакетов обслуживание запросов временных штампов в протоколе ICMP.

К настоящему моменту обеспечено полное покрытие тестами ядра ОС и библиотеки Си-функций. Общий объем существующих тестов составляет около 60 тыс. строк на языке Си. Для полного покрытия остальных компонент ОС разрабатывается еще около 60 тыс. строк тестов (табл. 2).

Аппаратно-зависимая часть	10 000 строк (ассемблер)
Ядро операционной системы	20 000 строк (Си)
Файловая система VFAT	8 000 строк (Си)
Поддержка сети	20 000 строк (Си)
Поддержка протокола TCP/IP	19 000 строк (Си)
Библиотека Си-функций	20 000 строк (Си)
Общее количество строк:	10 000 строк (ассемблер)
	87 000 строк (Си)

Таблица 2: Структура ос2000.

Реализация сетевых сервисов безопасности на платформе ос2000 и отечественной аппаратной платформы таких, как межсетевые экраны, сканеры безопасности и т. д., позволит получить решение, допускающее сквозную сертификацию по требованиям безопасности — от аппаратной платформы до прикладного уровня (рис. 2).

Отметим, что информационная безопасность сервисов на других платформах является в значительной степени вопросом веры, в лучшем случае, подкрепленной результатами тестирования.

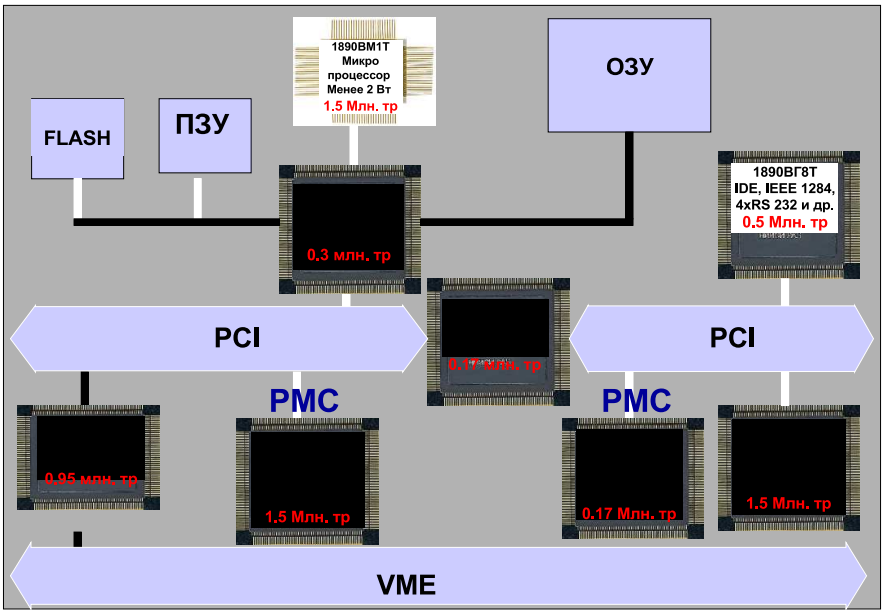


Рис. 2:

4.3 Отечественная аппаратная платформа

Основу аппаратной платформы составляет семейство магистрально-модульных ЭВМ в стандарте VME. Каждая ЭВМ, в общем случае, представляет собой набор слабо связанных по шине VME процессорных и дополнительных модулей в стандарте ЕВРОМЕХАНИКА-6U, размещенных в стандартном же механическом крейте, содержащем источник питания. Внутримодульной шиной обмена данными является PCI. На процессорные модули могут устанавливаться PMC мезонины с контроллерами SCSI, Ethernet, МКИО, IDE, графическим контроллером и т. д.

Процессорные и дополнительные модули реализованы на микропроцессорах и СБИС отечественной разработки, единичная сложность которых (рис. 2) не выше 1.5 млн. транзисторов. Наличие полностью отечественных проектов и относительно невысокая их сложность гарантируют возможность как формального, так и неформального доказательства их безопасности.

Литература

[1] Бетелин В. Б., Галатенко В. А., Годунов А. Н., Грюнталь А. И. Анализ информационной безопасности систем на платформе ОС РВ Багет. Безопасность информационных технологий, 2002, № 4.

[2] Бетелин В. Б., Галатенко В. А., Годунов А. Н., Грюнталь А. И. Обеспечение информационной безопасности систем на программной платформе ос2000. Настоящий сборник трудов, с. ?–?.

[3] Corn David. Working with Win32: the Good, the Bad, and the Ugly. Сетевой журнал «;login», November 1997 (русский перевод: `http://www.citforum.ru/operating_systems/articles/korn.shtml`).

Построение защищенных систем — глобальная стратегия Microsoft

В. Н. Мамыкин

Корпорация Microsoft своей стратегической и самой приоритетной задачей на ближайшие годы продолжает считать создание защищенных информационных систем. Анонсированная в 2002 году концепция «Защищенные информационные системы» (Trustworthy Computing, более подробно на русском языке см. `www.microsoft.com/rus/security`) представляет собой взгляд корпорации Microsoft на вопросы обеспечения информационной безопасности, включая задачи, стоящие как перед ИТ-индустрией, так и перед обществом в целом. Разработанный долгосрочный план реализации этой концепции продолжает выполняться. За это время проделана большая работа.

Как вы знаете, в октябре 2002 г. платформа Microsoft® Windows® 2000 получила международный сертификат Общие Критерии (Common Criteria) по классу EAL4+, который является самым высоким стандартом оценки степени защищенности операционных систем. Ни одна из других операционных систем не смогла получить еще такой высокой оценки защищенности до настоящего времени. Начата международная сертификация более современных систем Windows XP и Windows Server 2003 на соответствие уровню EAL4+ этого стандарта.

Значительной вехой в области обеспечения информационной безопасности стало подписание в январе 2003 года соглашения об использовании исходного кода в интересах повышения доверия к программным продуктам корпорации Microsoft, применяемым в органах государственной власти Российской Федерации. Договоренности, достигнутые корпорацией Microsoft, ФГУП НТЦ «Атлас» и ГУИР ФАПСИ в рамках данного соглашения, создают уникальные предпосылки для долговременного сотрудничества и использования современных технологий Microsoft в защищенных информационных системах органов государственной власти России. На базе НТЦ «Атлас» открыт комплекс по исследованию исходных кодов продуктов Microsoft, которым могут пользоваться представители всех заинтересованных государственных структур: ФСБ (ФАПСИ), Гостехкомиссии, Министерства обороны, Министерства атомной промышленности. Во время посещения этого комплекса представители всех этих структур наблюдали реальную работу с исходными кодами представителей спецслужб России и дали высокую оценку организации работ, позволяющей проводить автономные исследования кодов по закрытым методикам, не разглашая их даже представителям других служб. Эти работы будут продолжены в самое ближайшее время после передачи полномочий по сохранности кодов от ФАПСИ, подписавшей Соглашение о предоставлении кодов, к другим государственным организациям.

Выпущен Windows Server 2003 — первая серверная операционная система Microsoft, разработанная в рамках концепции защищенных информационных систем. Другим продуктом, выпущенным в рамках реализации концепции защищенных информационных систем, является Exchange Server 2003, в котором одной из самых перспективных технологий обеспечения безопасности является Управление цифровыми правами создаваемых документов (электронных писем, документов Word, Excel, Power Point и других). Эта технология, основанная на цифровых сертификатах, позволяет контролировать защищенность документа за пределами вашей корпоративной сети. Она удостоена диплома «Технология года» на осенней сессии конференции «Информационная безопасность России».

Важным шагом развития сотрудничества по построению защищенных систем явилось то, что операционные системы Windows Server 2003 и Windows XP переданы в Государственную техническую комиссию при Президенте Российской Федерации на сертификацию на соответствие российским требованиям по защите от несанкционированного доступа к информации. Мы рады продолжению нашего успешного сотрудничества с Гостехкомиссией России, с которой четыре года назад мы провели сертификацию операционных систем и СУБД Microsoft.

В настоящее время нами ведутся переговоры о сертификации наших продуктов с представителями и других основных сертифицирующих организаций России: ФСБ (ФАПСИ), Министерства обороны, Министерства атомной промышленности.

Нам приятно осознавать, что, являясь лидерами построения защищенных систем, мы уже давно

используем в наших продуктах такие технологии защищенности, как:

- Штатное использование для шифрования и электронной цифровой подписи (ЭЦП) сертифицированных российских криптоалгоритмов, разработанных нашими российскими партнерами в соответствии с международными стандартами совместимости.
- Проверка за счет использования ЭЦП (начиная с Windows 2000 Server) модулей операционной системы перед их запуском на предмет их искажения или подмены.
- Использование внутренней инфраструктуры открытых ключей (PKI) для всех работающих приложений.
- Работа с сертификатами открытых ключей формата X.509 на уровне ядра операционной системы (начиная с Windows 2000 Server), а не на уровне приложений.

Тем не менее перед мировым научным сообществом стоят еще многие задачи, сформулированные в концепции Защищенных информационных систем, которые необходимо решить. Наиболее актуальными из них для математиков можно считать:

- Разработка теории поведения сверхкрупных систем, построенных на основе слабых связей[1].
- Описание межмашинных процессов в модели peer-to-peer[2] с асинхронными компонентами сети.
- Описание моделей и связанных протоколов идентификации личности в информационных системах.
- Разработка новых средств программирования и автоматизированных средств анализа созданного программного обеспечения.
- Разработка моделей обеспечения функциональной совместимости защищенных систем, построенных на основе нефиксированной архитектуры и нефиксированной организации.

От решения этих и других научных задач зависит то, как быстро мы сможем все вместе продвинуться в вопросах построения действительно защищенных информационных систем следующего поколения, которым мы будем доверять так же, как доверяем сегодня электричеству и телефону.

[1] Слабые связи (loose affiliations, loose coupling) — ситуация, когда один компонент системы может делать минимум неявных допущений о характере работы другого компонента, с которым он взаимодействует. Например, если в рамках одного компьютера можно исходить из того, что другой компонент функционирует и без проблем получит отправленное ему сообщение («тесное связывание», tight coupling), то в распределенной сети удаленный компонент может оказаться отключен или перегружен. Чем меньше делается допущений о готовности к работе или об устройстве удаленного компонента, тем более «слабосвязанной» (loosely coupled) является система.

[2] Peer-to-peer — модель обмена информацией, когда устройства, подключенные к сети, могут напрямую связываться друг с другом, и каждое может выступать как в качестве клиента, так и в качестве сервера.

Некоторые проблемы вероятностной комбинаторики

Г. И. Ивченко, Ю. И. Медведев, В. Н. Сачков

I. Работа посвящена современным проблемам *вероятностной комбинаторики* (ВК). ВК — это одна из наиболее обширных и активно развивающихся областей дискретной математики, в которой вероятностными методами исследуются различные свойства комбинаторных объектов: подстановок, или взаимно однозначных отображений конечного множества в себя, однозначных отображений конечного множества в себя, графов, многочленов над конечными полями, разбиений конечных множеств и целых чисел и т. д. Каждое из перечисленных наименований представляет собой самостоятельное направление исследований в ВК со своей историей, многочисленными результатами и специфическими приложениями, в том числе к проблемам криптографии и защиты информации.

Начиная с классической работы В. Л. Гончарова «Из области комбинаторики» [1], в этом направлении получено много глубоких, интересных и важных для приложений результатов, опубликовано огромное количество работ, поток которых не прекращается. В данной работе мы ограничимся лишь кратким напоминанием основных положений и последних результатов этой теории, а основное внимание уделим новым подходам в этой проблематике.

II. Мы рассматриваем такие комбинаторные объекты, которые обладают свойством *разложимости* на отдельные компоненты: для подстановок — это *циклы*, для отображений конечного множества в себя и для графов — это *компоненты связности*, для многочленов над конечным полем — это *неприводимые сомножители* канонического разложения многочлена, для разбиений множеств — это *блоки* разбиения и т. д.

Пусть n есть характеристический параметр рассматриваемых объектов, их «вес» (для отображений и разбиений конечного множества n — число элементов, для графов n — число вершин, для многочленов n — степень и т. д.). Символом $\mathcal{K}_n$ обозначим множество некоторых разложимых объектов веса n и для конкретного объекта $K \in \mathcal{K}_n$ пусть $c_i(n)$ обозначает число его компонент веса i $(i = 1, 2, \ldots, n)$. Набор $c(n) = (c_1(n), \ldots, c_n(n))$ называется *структурой* объекта K. Наконец, пусть $N(n, c(n))$ обозначает число объектов из $\mathcal{K}_n$ со структурой $c(n)$.

Суть вероятностного подхода при изучении структурных свойств разложимых комбинаторых объектов состоит в том, что на множестве $\mathcal{K}_n$ тем или иным способом вводится вероятностная мера P, приписывающая каждому объекту $K \in \mathcal{K}_n$ вероятность его наблюдения $\mathsf{P}(K)$ — так возникают *случайные комбинаторные объекты*: случайные подстановки, случайные отображения, случайные графы, случайные многочлены над конечным полем, случайные разбиения множеств и чисел и т. д. Для случайного объекта его структура $c(n)$ становится случайной величиной, для изучения свойств которой применяются различные вероятностные методы и особенно эффективно — предельные теоремы теории вероятностей, которые позволяют исследовать асимптотические особенности изучаемых объектов при $n \to \infty$.

До настоящего времени большинство исследований в ВК ограничивалось рассмотрением лишь *равновероятных* объектов, т. е. когда P — равномерная мера на множестве $\mathcal{K}_n$ либо на некотором выделенном его подмножестве. Такой выбор меры естественен и удобен для решения различных *перечислительных* задач комбинаторики, в которых исследуется число объектов из $\mathcal{K}_n$ с теми или иными структурными свойствами. Основной вклад в развитие этого направления ВК внесли В. Л. Гончаров, В. Е. Степанов, В. Н. Сачков, В. Ф. Колчин, Харди, Пойа, Макмагон, Рота, Эрдёш, Реньи и др. Обзор и библиографию соответствующих результатов см. в [2]–[14].

Внутренняя логика развития теории и современные приложения порождают стремление перейти от рассмотрения равновероятных объектов к более общим моделям случайных объектов, допускающих те или иные отклонения от равновероятности. Но на сегодня такая сколь-нибудь общая теория отсутствует; имеются лишь отдельные (но весьма интересные) результаты, о чем будет сказано ниже.

В настоящее время имеется возможность рассмотреть новый подход в этой проблематике, связанный с введением на произвольном множестве разложимых объектов $\mathcal{K}_n$ некоторой общей параметри-

ческой меры, обладающей достаточным числом степеней свободы, чтобы удовлетворить потребности практики в рассмотрении неравновероятных комбинаторных объектов самой различной природы [15].

III. Пусть $\mathcal{K}_n$ — произвольное множество разложимых кобинаторных объектов веса n. Зададим на этом множестве вероятностную меру, зависящую от параметра $\theta = (\theta_1, \ldots, \theta_n)$, $\theta_i \geqslant 0$, в соответствии с которой произвольный объект $K \in \mathcal{K}_n$ со структурой $c(n) = a = (a_1, \ldots, a_n)$ наблюдается с вероятностью, пропорциональной $\prod_i \theta_i^{a_i}$. Именно,

$$\mathsf{P}_\theta(K) = I\left(\sum_{i=1}^n i a_i = n\right) \cdot \prod_{i=1}^n \theta_i^{a_i} \Big/ H_n(\theta), \tag{1}$$

где $I(\cdot)$ — индикатор и $H_n(\theta)$ — необходимый нормирующий множитель, определяемый условием

$$\sum_{K \in \mathcal{K}_n} \mathsf{P}_\theta(K) = 1.$$

Тогда

$$H_n(\theta) = \sum_{a: \sum_i i a_i = n} N(n, a) \prod_{i=1}^n \theta_i^{a_i}, \tag{2}$$

т. е. это есть производящая функция чисел $N(n, a)$ для множества рассматриваемых объектов $\mathcal{K}_n$, которую будем называть *статистической суммой* множества $\mathcal{K}_n$.

В модели (1) структура случайного объекта имеет распределение

$$\mathsf{P}_\theta(c(n) = a) = I\left(\sum_{i=1}^n i a_i = n\right) N(n, a) \prod_{i=1}^n \theta_i^{a_i} \Big/ H_n(\theta) \tag{3}$$

и для ее производящей функции имеет место представление

$$F_{n\theta}(t) = \mathsf{E}_\theta \prod_{i=1}^n t_i^{c_i(n)} = H_n(t \cdot \theta)/H_n(\theta), \tag{4}$$

где $t \cdot \theta = (t_1\theta_1, t_2\theta_2, \ldots, t_n\theta_n)$.

Соотношение (4) может быть основой для решения самых различных задач вероятностной комбинаторики. Выбирая соответствующим образом значения «управляемого» параметра $\theta = (\theta_1, \ldots, \theta_n)$, мы можем задавать на множестве $\mathcal{K}_n$ различные распределения, позволяющие исследовать различные особенности рассматриваемых комбинаторных объектов, решать для них не только стандартные перечислительные задачи, но и проводить их более глубокий вероятностный и статистический анализ.

IV. Отметим некоторые частные случаи модели (1), рассматривавшиеся ранее в литературе.

1. Если все $\theta_i = 1$, то имеем равномерную меру на множестве $\mathcal{K}_n$, приписывающую каждому объекту $K \in \mathcal{K}_n$ одну и ту же вероятность $\mathsf{P}(K) = |\mathcal{K}_n|^{-1}$, где $|\mathcal{K}_n|$ — общее число объектов множества $\mathcal{K}_n$. В этом случае формула (3) принимает вид

$$\mathsf{P}_\theta(c(n) = a) = I\left(\sum_{i=1}^n i a_i = n\right) \frac{N(n, a)}{|\mathcal{K}_n|}. \tag{5}$$

Это соотношение лежит в основе всех перечислительных задач комбинаторики и большинство результатов ВК получено именно в рамках модели (5).

2. Пусть для заданного подмножества $A \subseteq \{1, 2, \ldots, n\}$ символ $\mathcal{K}_n(A)$ обозначает подмножество тех объектов $K \in \mathcal{K}_n$, веса компонент которых принадлежат A. Чтобы изучать различные структурные свойства объектов из подмножества $\mathcal{K}_n(A)$, следует в модели (1) положить $\theta_i = 0$ при $i \notin A$. Если дополнительно положить $\theta_i = 1$ при $i \in A$, то получим равномерную меру на подмножестве $\mathcal{K}_n(A)$, приписывающую каждому объекту $K \in \mathcal{K}_n(A)$ вероятность $|\mathcal{K}_n(A)|^{-1}$. Примерами задания подмножества A являются: $A = \{i : i \leqslant s\}$ при некотором $s < n$ (размеры компонент не превосходят s), либо $A = \{i : i \equiv m \pmod d\}$ при некоторых m и d и т. д.

а) Так в [11] рассматривались так называемые $A\Lambda$-разбиения конечного n-множества, для которых веса компонент (блоков) и частота их встречаемости в разбиении являются элементами некоторых

последовательностей A и Λ. На множестве $A\Lambda$-разбиений задается равномерное вероятностное распределение. К реализации некоторого $A\Lambda$-разбиения в результате случайного испытания применяется другой случайный процесс нанесения меток для блоков.

Блок веса k получает метку с вероятностью p_k и остается непомеченным с вероятностью $q_k = 1-p_k$.

Если $\xi^{(k)}$ — число помеченных k-блоков в случайном $A\Lambda$-разбиении n-множества, $f(x_1,\ldots,x_n;A,\Lambda)$ — производящая функция для $\xi_n(A,\Lambda) = \left(\xi^{(1)},\ldots,\xi^{(n)}\right)$, $\Lambda = (\Lambda_1,\ldots,\Lambda_n)$ и $T_n(A,\Lambda)$ — число $A\Lambda$-разбиений n-множества, то

$$\sum_{n=0}^{\infty} T_n(A,\Lambda)\cdot f(x_1,\ldots,x_n;A,\Lambda)\frac{t^n}{n!} = \prod_{j\in A}\sum_{\beta_j\in\Lambda_j}\left((q_j+p_jx_j)\frac{t^j}{j!}\right)^{\beta_j}\frac{1}{\beta_j!}.$$

Отсюда, в частности, при $A=(1,2,\ldots)$, $\Lambda=(0,1,2,\ldots)$ следует, что

$$\sum_{n=0}^{\infty} T_n\cdot f(x_1,\ldots,x_n)\frac{t^n}{n!} = \exp\left\{e^t-1+\sum_{k=1}^{\infty}p_k(x_k-1)\frac{t^k}{k!}\right\},$$

где T_n — числа Бэлла.

Случайные величины $\xi(n)$ и η_n, равные числу помеченных блоков и числу элементов, содержащихся в помеченных блоках, равны

$$\xi(n) = \xi^{(1)}+\xi^{(2)}+\ldots+\xi^{(n)},$$
$$\eta_n = 1\cdot\xi^{(1)}+2\cdot\xi^{(2)}+\ldots+n\xi^{(n)}.$$

При $n\to\infty$ имеют место следующие утверждения:

1. Если $p_j(\lambda/r)^{-j}\to 1$ при $j=1,2,\ldots$, и $re^r=n$, $0\leqslant\lambda<\infty$, то $\mathcal{L}(\xi(n))\to\Pi(e^\lambda-1)$ ($\Pi(\lambda)$ — распределение Пуассона с параметром λ).

2. Если $p_j=\lambda j/n$, $j=1,2,\ldots,n$, $0<\lambda<\infty$, то
$$\mathcal{L}(\xi(n))\to\Pi(\lambda).$$

3. Если $p_j(\lambda/r)^{-j}\to 1$, $j=1,\ldots,n$, $re^r=n$, $0<\lambda<\infty$, $0<l<\infty$, то
$$\mathcal{L}\left(\xi^{(l)}\right)\to\Pi\left(\frac{\lambda^l}{l!}\right).$$

4. Если $p_j=\delta^j$, $j=1,\ldots,n$, $\delta=\delta(n)$, $r\delta\to\lambda<\infty$, то
$$\mathsf{P}(\eta_n=k)\to\frac{\lambda^k}{k!}\cdot T_k\cdot e^{-(e^\lambda-1)},\quad k=0,1,\ldots.$$

5. Если $p_j=\delta^j$, $j=1,2,\ldots,n$, $0<\delta<1$, $re^r=n$, то
$$\mathcal{L}\left(\frac{\xi(n)-e^{\delta r}}{e^{\delta r/2}}\right)\to\mathcal{N}(0,1),$$
где $\mathcal{N}(0,1)$ — стандартное нормальное распределение.

б) Асимптотическая нормальность числа компонент случайных подстановок имеет место, если задать равномерную меру на подмножествах подстановок $S_n^{(1)}$ или $S_n^{(2)}$, содержащих циклы лишь четной или нечетной длины, на подмножестве перемещенных подстановок (без единичных циклов), на подмножествах, в которых конечное число некоторых длин циклов запрещены, на подмножествах, содержащих лишь циклы, большие некоторого фиксированного числа s и т. д.

Эти примеры, а также другие примеры такого задания равномерной меры на подмножествах комбинаторых объектов можно найти в [2, 3, 4, 5], а также в [16, 17, 18] и в библиографии к этим работам.

3. Если в (1) положить $\theta_1 = \theta_2 = \ldots = \theta_n = \theta > 0$, то получим однопараметрическую меру, приписывающую объектам $K \in \mathcal{K}_n$ вероятности, пропорциональные $\theta^{|K|}$, где $|K| = c_1(n) + \ldots + c_n(n)$ есть общее число компонент объекта K. Впервые такая неравновероятная мера рассматривалась Эвансом для случайных подстановок [19]. Модель Эванса и ее аналоги для других комбинаторных объектов были предметом рассмотрения в работах [9], [20, 21, 22]. Так, производящая функция F_θ цикловой последовательности случайных подстановок $c_\theta(n) = (c_{\theta 1}(n), \ldots, c_{\theta n}(n))$ имеет вид [22]

$$F_\theta(t_1, \ldots, t_n) = \frac{n!}{\theta_{(n)}}[z^n]\frac{1}{(1-z)^\theta}\exp\left\{\theta\sum_{i=1}^{\infty}\frac{z^i}{i}(t_i - 1)\right\}$$
$$(\theta_{(n)} = \theta(\theta+1)\ldots(\theta+n-1), \quad [z^n] \equiv \mathrm{coef}_{z^n}).$$

Отсюда стандартными методами можно получить как интегральный, так и локальный вариант центральной предельной теоремы с оценкой скорости сходимости для числа циклов $|c_\theta(n)| = \sum_i c_{\theta i}(n)$ при $\mathsf{D}|c_\theta(n)| \to \infty$, а также пуассоновскую сходимость в схеме серий, т. е. когда $\theta = \theta(n)$ стремится к 0 либо к ∞ и $\mathsf{D}|c_\theta(n)| \to \infty$ стремится к постоянной. Также можно получить формулы для нахождения предельных при $n \to \infty$ распределений членов вариационного ряда, составленного из длин циклов подстановки, а также других характеристик случайных подстановок.

V. Эффективность использования общего представления (4) зависит, конечно, от сложности явного выражения статистической суммы $H_n(\theta)$, определенной в (2). В комбинаторике известны три общих типа объектов, для которых статистическая сумма может быть выписана в явной форме: это *ансамбли*, *мультимножества* и *селекции*.

Термин ансамбли объединяет, в частности, подстановки, однозначные отображения множества $X_n = \{1, 2, \ldots, n\}$ в себя, разбиения множества X_n, графы. Если m_i есть число соответствующих объектов веса i, состоящих из единственной компоненты, то формула для чисел $N(n, a) = N^{(1)}(n, a)$ имеет вид

$$N^{(1)}(n, a) = I\left(\sum_{i=1}^{n} i a_i = n\right) \cdot n! \prod_{i=1}^{n}\left(\frac{m_i}{i!}\right)^{a_i}\frac{1}{a_i!}, \tag{6}$$

а статистическая сумма есть

$$H_n^{(1)}(\theta) = [z^n]n!\exp\left\{\sum_{i=1}^{n}\frac{z^i}{i!}m_i\theta_i\right\}. \tag{7}$$

Числа m_i для ряда конкретных объектов известны:

- для подстановок $m_i = (i-1)!$;
- для однозначных отображений $m_i = (i-1)!\sum_{j=0}^{i-1}(i^j/j!)$;
- для нормированных (т. е. со старшим коэффициентом 1) многочленов над полем $\mathrm{GF}(q)$ (q — простое или степень простого числа) $m_i = (1/i)\sum_{r|i}\mu(r)q^{i/r}$, $\mu(r)$ — функция Мебиуса;
- для разбиений конечного множества $m_i = 1\ \forall\, i$.

Мультимножества, или неупорядоченные выборки с повторениями элементов, включают, в частности, нормированные многочлены над конечным полем и разбиения целого числа на слагаемые. В этом случае

$$N(n, a) = N^{(2)}(n, a) = I\left(\sum_{i=1}^{n} i a_i = n\right) \cdot \prod_{i=1}^{n}\binom{m_i + a_i - 1}{a_i} \tag{8}$$

и

$$H_n^{(2)}(\theta) = [z^n]\prod_{i=1}^{n}(1 - z^i\theta_i)^{-m_i}. \tag{9}$$

Наконец, селекции, или неупорядоченные выборки без повторения элементов, включают, в частности, разбиения целого числа на различные слагаемые и свободные от квадратов многочлены. Для таких объектов

$$N(n, a) = N^{(3)}(n, a) = I\left(\sum_{i=1}^{n} i a_i = n\right) \cdot \prod_{i=1}^{n}\binom{m_i}{a_i} \tag{10}$$

и

$$H_n^{(3)}(\theta) = [z^n] \prod_{i=1}^{n} (1 + z^i \theta_i)^{m_i}. \tag{11}$$

В итоге можно констатировать, что для широкого круга комбинаторных объектов соотношения (4) и (7), (9), (11) дают общий математический аппарат исследования их различных вероятностно-статистических свойств.

В качестве примера использования представления (4) приведем общую формулу для смешанных факториальных моментов случайной структуры $c(n)$ в случае ансамблей:

$$\mathsf{E}_\theta \prod_i (c_i(n))_{r_i} = \frac{n!}{(n-m)!} \frac{H_{n-m}^{(1)}(\theta)}{H_n^{(1)}(\theta)} \prod_i \left(\frac{m_i \theta_i}{i!} \right)^{r_i} \tag{12}$$

(здесь $m = \sum_i i r_i$, $(x)_r = x(x-1)\dots(x-r+1)$).

Литература

[1] Гончаров В. Л. Из области комбинаторики. Изв. АН СССР, сер. матем., 1944, 8, № 1, 3–48.

[2] Сачков В. Н. Введение в комбинаторные методы дискретной математики. М.: Наука, 1982.

[3] Сачков В. Н. Комбинаторные методы дискретной математики. М.: Наука, 1977.

[4] Колчин В. Ф. Случайные отображения. М.: Наука, 1984.

[5] Колчин В. Ф. Случайные графы. М.: Физматлит, 2000.

[6] Степанов В. Е. О вероятности связности случайного графа $g_m(t)$. Теория вероятн. и ее примен., 1970, 15, № 1, 55–67.

[7] Степанов В. Е. Предельные распределения некоторых характеристик случайных отображений. Теория вероятн. и ее примен., 1969, 14, № 4, 639–653.

[8] Erdős P., Renyi A. On the evolution of random graphs. Publ. Math. Inst. Hungarian Acad. Sci., Ser. A., 1960, 5, 17–61.

[9] Arratia R., Barbour A. D., Tavare S. On Poisson–Dirichlet limits for random decomposable combinatorial structures. Comb. Probab. Comp., 1999, 8, 193–208.

[10] Arratia R., Tavare S. Independent process approximations for random combinatorial structures. Adv. Math., 1994, 104, 90–154.

[11] Сачков В. Н. Случайные разбиения множеств. Математические вопросы кибернетики, М.: Наука, 1999, 6, 33–54.

[12] Ивченко Г. И., Медведев Ю. И. О структуре случайных многочленов над конечными полями. Труды по дискретной математике, 2000, 3, 111–138.

[13] Ивченко Г. И., Медведев Ю. И. Случайные многочлены над конечным полем. Теория вероятн. и ее применен., 1996, 41, № 1, 204–210.

[14] Ивченко Г. И., Медведев Ю. И. О случайных подстановках. Труды по дискретной математике, 2002, 5, 73–92.

[15] Ивченко Г. И., Медведев Ю. И. Неравновероятные меры на множествах разложимых комбинаторных объектов. Обозрение прикл. промышл. матем., сер. дискретн. матем., 2003, 10, № 2, 348–349.

[16] Сачков В. Н. Отображения конечного множества с ограничениями на контуры и высоту. Теория вероятн. и ее примен., 1972, 17, № 4, 679–694.

[17] Болотников Ю. В., Сачков В. Н., Тараканов В. Е. Асимптотическая нормальность некоторых величин, связанных с цикловой структурой случайных подстановок. Матем. сб., 1976, 99, № 1, 121–133.

[18] Грушо А. А. Случайные отображения ограниченной кратности. Теория вероятн. и ее примен., 1972, 17, № 3, 440–449.

[19] Ewens W. J. The sampling theory of selectively neutral alleles. Theoret. Population Biol., 1972, 3, 87–112.

[20] Hansen J. C. A functional central limit theorem for the Ewens Sampling Formula. J. Appl. Probab., 1990, 27, 28–43.

[21] Arratia R., Barbour A. D., Tavare S. Poisson process approximations for the Ewens Sampling Formula. Ann. Appl. Probab., 1992, 2, 519–535.

[22] Ивченко Г. И., Медведев Ю. И. Метод В. Л. Гончарова и его развитие в анализе различных моделей случайных подстановок. Теория вероятн. и ее примен., 2002, 47, № 3, 558–566.

Случайные системы уравнений и их криптографические приложения

Г. В. Балакин

1 Основные понятия и задачи анализа

1.1 Введение

Как известно, в криптографии открытый текст преобразуется в шифрованный текст с помощью так называемых уравнений шифрования. Поэтому естественен интерес криптографов к анализу и решению систем уравнений над различными алгебраическими структурами.

Системы таких уравнений появляются также в задачах комбинаторики, теории графов, теории кодирования, математической экономики, анализа дискретных автоматов, теории классификации и распознавания образов, целочисленного программирования и других областях дискретной математики. Но системы уравнений, возникающие в криптографии, обладают некоторыми особенностями, на которые в дальнейшем изложении мы будем обращать внимание.

Анализу и методам решения систем уравнений в дискретной области посвящено очень много литературы. Среди исследователей отметим докторов физико-математических наук Степанова В. Е., Коваленко И. Н., Балакина Г. В., Колчина В. Ф., Левитскую А. А., Копытцева В. А., Севастьянова Б. А., Сачкова В. Н., Зубкова А. М., Михайлова В. Г., внесших вклад в развитие теории случайных систем уравнений, связанных с криптографией.

Целью настоящей работы является ознакомление с некоторыми основными результатами и направлениями исследований в этой области. Автор не претендует на полный обзор всех результатов и полный список использованной литературы. Заинтересованный читатель может найти необходимые ему сведения в приведенном кратком списке литературы, где приведены более полно результаты исследований и соответствующая литература.

1.2 Классификация систем уравнений

Рассмотрим систему уравнений

$$f_i(x_1, \ldots, x_n) = a_i, \quad i = 1, \ldots, t, \tag{1}$$

в которой

$$x_i \in N, \quad |N| = q, \quad a_i \in M, \quad |M| = m, \quad i = 1, \ldots, t,$$

N, M— некоторые множества, например, $N = \{0, 1, \ldots, q-1\}$, $M = \{0, 1, \ldots, m-1\}$. Каждая функция f_i $(i = 1, \ldots, t)$ задает отображение N^n в M.

Если система (1) рассматривается над конечным полем $\mathrm{GF}(q)$, то $m = q$ и элементы поля для удобства будем отождествлять с наименьшими неотрицательными вычетами по модулю q. При $m = q = 2$ система (1) называется *булевой*. Особое промежуточное положение занимают так называемые *псевдобулевы* системы, для которых неизвестные $x_1, \ldots, x_n$ булевы, т. е. принимают только два значения 0 и 1, а функции $f_1, \ldots, f_t$ задают отображения булевых векторов в поле вещественных чисел R. В криптографии неизвестные $x_1, \ldots, x_n$ — элементы ключа, а система уравнений (1) называется *системой уравнений гаммообразования.*

Одним из подходов к анализу свойств систем уравнений вида (1) является задание вероятностной меры на множестве таких систем и изучение получающихся при этом случайных характеристик систем, например, среднего числа решений, структуры решений, распределения числа решений, числа уравнений, необходимого для однозначного определения решения системы либо для ее несовместности

(отсутствия решений). Характеристики, усредненные по всему рассматриваемому классу систем уравнений, дают предварительные ориентиры при анализе любой конкретной системы из данного класса систем. Заметим, что в некоторых случаях случайность системы уравнений естественно порождается физической природой рассматриваемой задачи.

Вначале выделим важный класс случайных систем уравнений. В общем случае случайная система уравнений

$$\varphi_i(x_1,\dots,x_n)=c_i,\quad i=1,\dots,t \tag{2}$$

по некоторому вероятностному закону выбирается из некоторой совокупности систем уравнений $\Omega=\{\omega\}$. Здесь под элементарным событием $\omega=\left(\overline{f},\overline{a}\right)$ мы подразумеваем конкретную реализацию системы уравнений (2):

$$\left\{\overline{\varphi}=(\varphi_1,\dots,\varphi_t)=\overline{f}=(f_1,\dots,f_t),\quad \overline{c}=(c_1,\dots,c_t)=\overline{a}=(a_1,\dots,a_t)\right\}.$$

Каждому элементарному событию приписывается вероятность

$$p(\omega)>0,\quad \sum_{\omega\in\Omega}p(\omega)=1.$$

Классификацию систем уравнений можно проводить по трем параметрам:

1) характер зависимости правых и левых частей в системе,

2) алгебраический вид левых частей,

3) способ задания правых частей.

По характеру зависимости правых и левых частей системы уравнений могут быть полуслучайными, случайными и заведомо совместными случайными. Приведем соответствующие определения.

Определение 1. Система уравнений называется *полуслучайной*, если либо её левая часть случайна, т. е. выбирается по некоторому вероятностному закону, а правая часть фиксирована, либо её правая часть случайна, а левая часть фиксирована.

Определение 2. Система (2) называется *случайной* системой уравнений с независимыми частями, если её левая часть $\overline{\varphi}=(\varphi_1,\dots,\varphi_t)$ не зависит от правой части $\overline{c}=(c_1,\dots,c_t)$.

По сути дела, введенные в определениях 1 и 2 понятия, относятся не столько к вероятностной структуре случайной системы, сколько к форме ее записи. Например, в любой системе можно перенести правые части в левые, сделав её полуслучайной. Но в криптографии разделение правых и левых частей является существенным. Действительно, левая часть зависит от ключа, а правая часть, как правило, есть шифрованный текст либо сумма открытого и шифрованного текстов.

Для случайной системы уравнений с независимыми частями выполняется равенство

$$p(\omega)=\widehat{p}\left(\overline{f}\right)\widetilde{p}(\overline{a}),\quad \omega=\left(\overline{f},\overline{a}\right)\in\Omega,$$

где $\widehat{p}$ и $\widetilde{p}$ — вероятностные меры на множествах значений левой и правой частей случайной системы уравнений, соответственно. Обозначим ξ_t число решений случайной системы из t уравнений.

Выделим ещё один важный для криптографии класс случайных систем уравнений. Пусть в системе (1)

$$f_i(\overline{x})=a_i,\quad a_i=f_i\left(\overline{x}^0\right),\quad i=1,\dots,t \tag{3}$$

где $\overline{x}^0=(x_1^0,\dots,x_n^0)$ есть некоторый вектор, называемый обычно истинным решением. Эта система отличается от системы (2) тем, что правые части системы (3) всегда (при любом числе уравнений) удовлетворяют условиям $a_i=f_i\left(\overline{x}^0\right)$, $i\geqslant 1$.

Определение 3. Система уравнений (2) называется *заведомо совместной* системой уравнений, если

$$\mathsf{P}\left\{\overline{c}\in\bigcup_{\overline{x}}\{\overline{\varphi}(\overline{x})\}\right\}=1.$$

Таким образом, истинное решение удовлетворяет системе (3) при любом её расширении (добавлении уравнений). Ложное решение может не удовлетворить некоторому уравнению при расширении системы. Обозначим через η_t число решений заведомо совместной случайной системы из t уравнений. Очевидно, что $\eta_t \geqslant 1$.

В заведомо совместную систему случайность может быть введена двумя способами. Первый способ, как и для случайных систем, связан со случайным выбором функций $f_1, \ldots, f_t$. В этом случае левая часть системы (3) случайная, а правая часть однозначно задается левой частью и вектором $\overline{x}^0$. При другом способе левая часть системы (3) фиксирована, а правая часть становится известной только после случайного выбора истинного решения $\overline{x}^0$. В обоих случаях правая часть случайная, так как зависит от выбора либо левой части, либо $\overline{x}^0$.

Системы уравнений различаются также по алгебраическому виду левых частей. Системы уравнений могут рассматриваться над конечным полем $\mathrm{GF}(q)$, кольцом, группой или над полем вещественных чисел. Системы уравнений над R обычно называются *целочисленными*. В частности, системы могут быть булевыми или псевдобулевыми. Возможны и смешанные системы уравнений, когда часть уравнений, к примеру, булевы (т. е. над $\mathrm{GF}(2)$), а часть уравнений псевдобулевы (т. е. над R). Кроме того, системы уравнений могут быть линейными и нелинейными, с фиксированным или случайным числом неизвестных в каждом уравнении, и т. д.

В заключение кратко скажем о наиболее типичных способах задания правых частей в системах уравнений. Системы могут иметь точно известную правую часть, правую часть, известную в вариантах, и искаженную правую часть. Заметим, что для заведомо совместной системы уравнений её истинная правая часть обязательно содержится среди возможных вариантов правых частей, а при наличии искажений система может оказаться несовместной. В последнем случае иногда в правую часть вводят так называемые мешающие параметры (см. [2]), характеризующие искажения, и система уравнений остается заведомо совместной, т. е. имеет истинное решение $\overline{x}^0$ при некоторых истинных значениях мешающих параметров (искажений).

1.3 Задачи анализа систем уравнений

Основная задача для случайных систем — определить или оценить величину $\mathsf{P}\{\xi_t > 0\}$ — вероятность совместности системы. Представляют интерес также среднее число решений и структура множества решений. Более сложная задача — найти распределение числа решений ξ_t.

Зная функциональную зависимость среднего числа решений случайной системы от числа уравнений, нетрудно оценить так называемый «порог единственности» — минимальное число уравнений t_0, при котором среднее число решений случайной системы не превосходит 1:

$$\mathsf{E}\xi_{t_0} \leqslant 1, \quad \mathsf{E}\xi_{t_0-1} > 1.$$

Очень важно для случайной системы оценить момент t^* — «порог несовместности», когда система становится несовместной:

$$\mathsf{E}\xi_{t^*-1} \geqslant 1, \quad \mathsf{E}\xi_{t^*} = 0.$$

Найти распределение этого момента t^*, как правило, очень трудно. Для линейных систем над полем $\mathrm{GF}(q)$ он связан с появлением линейных зависимостей между строками матрицы системы.

В работе Сачкова В. Н. [3] величина t^* названа индексом несовместности системы при постановке в неё фиксированного вектора $\overline{x}$ или индексом покрытия случайными множествами всего множества возможных решений. Для некоторых случаев задания случайных функций в левой части системы найдены предельные распределения этого индекса.

При изучении заведомо совместных систем основная задача — найти распределение числа решений и структуру множества решений. Более простыми задачами могут оказаться оценка среднего числа решений и вероятностных характеристик числа уравнений, при котором заведомо совместная система с заданной вероятностью имеет только одно решение. Для линейных заведомо совместных систем над полем $\mathrm{GF}(q)$ с матрицей A_t распределение числа решений определяется рангом $r(A_t)$ случайной матрицы A_t

$$\mathsf{P}\{\eta_t = q^{n-r(A_t)}\} = 1, \quad \mathsf{P}\{\eta_t = 1\} = \mathsf{P}\{r(A_t) = n\}.$$

По аналогии со случайными системами определим «порог единственности» ложных решений для заведомо совместной системы равенством

$$t_0 = \min\{t : \mathsf{E}(\eta_t - 1) \leqslant 1,\ \mathsf{E}(\eta_{t-1} - 1) > 1\},$$

т. е. t_0 — минимальное число уравнений, при котором среднее число ложных решений заведомо совместной системы не превосходит 1 и «порог однозначности»

$$t^* = \min\{t : \eta_{t-1} \geqslant 2,\ \eta_t = 1\},$$

т. е. t^* — минимальное число уравнений, при котором заведомо совместная система имеет единственное решение. Ложные решения при $t \geqslant t^*$ отсутствуют.

Предположим, что во всех t уравнениях заведомо совместной системы левые части не зависят от ν_t неизвестных. Тогда очевидно, что в заведомо совместной системе $\nu_t \geqslant q^{\nu_t}$. Здесь возникает, как правило, комбинаторная задача нахождения распределения случайной величины ν_t. Например, оценка такого момента t, что $\nu_{t-1} > 0$, а $\nu_t = 0$.

Очень существенным моментом для заведомо совместной системы является изучение структуры множества возможных решений, связи ложных решений с истинным решением. Например, если с вероятностью, стремящейся к 1 при $t, n \to \infty$, ложные решения отличаются от истинного решения $\overline{x}^0$ не более чем $k(n)$ координатами, $k(n) = o(n)$, то первое же найденное решение системы будет либо истинным, либо даст существенную информацию об истинном решении, которая может быть использована при поиске остальных решений системы.

Заметим, что при анализе числа решений, их структуры, связи с истинным решением обычно удается выделить такие события, которые в основном и определяют поведение отмеченных параметров и наличие свойств. При разработке эффективных методов решения такие результаты могут оказаться полезными и подсказать исследователю способ решения задачи.

1.4 Связь между распределениями чисел решений случайных и заведомо совместных случайных систем уравнений

Пусть в системе уравнений (2) неизвестный вектор $\overline{x}$ может принимать K возможных значений $\overline{x}(1), \ldots, \overline{x}(K)$, а значения $c_1, \ldots, c_t$ правой части $(\varphi_1, \ldots, \varphi_t)$ в одном случае выбираются из множества M^t независимо от левой части по некоторому вероятностному закону, а в другом случае есть результат подстановки в левую часть системы некоторого фиксированного значения $\overline{x} = \overline{x}(j)$. Распределения левых частей в обоих случаях одинаковы.

Пусть, далее, Ω_i — множество реализаций случайной системы (2), которым удовлетворяет значение $\overline{x} = \overline{x}(i)$, $i = 1, \ldots, K$; $p(\omega) = \widehat{p}\,(\overline{f})\,\widetilde{p}(\overline{a})$ есть вероятность конкретной реализации $\omega = (\overline{f}, \overline{a})$ случайной системы (2); $\Omega(j)$ — множество реализаций заведомо совместной системы (2) с истинным решением $\overline{x} = \overline{x}(j)$; $\eta(j)$ — число решений заведомо совместной системы с истинным решением $\overline{x}(j)$.

Определение 4. Назовём случайную систему уравнений с независимыми частями и заведомо совместную случайную систему уравнений вида (2) согласованными, если для любого $i = 1, \ldots, K$

1) $\Omega_i \equiv \Omega(i)$;

2) $\widehat{p}\,(\overline{f}) = p(\omega)/p(\Omega_i)$, $\omega = (\overline{f}, \overline{f}(\overline{x}(i)))$;

3) распределение $\eta(i)$ не зависит от номера i.

Лемма 1 ([1]). *Если правая часть случайной системы уравнений с независимыми частями* (2) *имеет равномерное распределение на множестве допустимых правых частей, то первое и второе условия согласованности выполняются.*

Теорема 1 ([1]). *Если случайная система уравнений с независимыми частями и заведомо совместная случайная система согласованны, то между распределениями их чисел решений ξ и η существует следующая связь:*

$$\begin{aligned} \mathsf{P}\{\xi = 0\} &= 1 - \mathsf{E}\xi \sum_{k \geqslant 1} \frac{\mathsf{P}\{\eta = k\}}{k}, \\ \mathsf{P}\{\xi = 0\} &= \mathsf{E}\xi \frac{\mathsf{P}\{\eta = k\}}{k}, \quad k \geqslant 1. \end{aligned} \tag{4}$$

Сформулируем утверждение, которое по существу является следствием теоремы 1.

Лемма 2 ([1]). *Пусть случайная система уравнений и заведомо совместная система уравнений согласованны. Тогда* $\mathsf{P}\{\xi \geqslant 1\}$ *(вероятность совместности случайной системы) и* $\mathsf{P}\{\eta \geqslant 1\}$ *(вероятность единственности решения заведомо совместной системы) удовлетворяют следующим неравенствам:*

$$\begin{gathered} \mathsf{E}\xi\mathsf{P}\{\eta = 1\} \leqslant \mathsf{P}\{\xi \geqslant 1\} \leqslant \mathsf{E}\xi[1 + \mathsf{P}\{\eta = 1\}]/2, \\ 12\mathsf{P}\{\xi \geqslant 1\} \leqslant \mathsf{E}\xi[4 + \mathsf{E}\eta + 7P\{\eta = 1\}], \\ \mathsf{P}\{\xi \geqslant 1\} \geqslant \mathsf{E}\xi/\mathsf{E}\eta. \end{gathered} \tag{5}$$

В заключение найдём условное распределение при условии $\xi \geqslant 1$. Очевидно, что

$$\begin{gathered} \mathsf{P}\{\xi = k \mid \xi \geqslant 1\} = \mathsf{P}\{\xi = k\}/[1 - \mathsf{P}\{\xi = 0\}], \quad k \geqslant 1, \\ \mathsf{E}(\xi \mid \xi \geqslant 1) = \mathsf{E}\xi/[1 - \mathsf{P}\{\xi = 0\}]. \end{gathered}$$

Поэтому из (4) следует, что

$$p_k = \mathsf{P}\{\xi = k \mid \xi \geqslant 1\} = \mathsf{E}(\xi \mid \xi \geqslant 1)\frac{\mathsf{P}\{\eta = k\}}{k}, \quad k \geqslant 1,$$

При $k < \mathsf{E}(\xi \mid \xi \geqslant 1)$ величина $p_k > \mathsf{P}\{\eta = k\}$, а при $k < \mathsf{E}(\xi \mid \xi \geqslant 1)$ величина $p_k < \mathsf{P}\{\eta = k\}$.

Следовательно, по числу решений можно статистически различать заведомо совместную случайную систему и случайную систему, имеющую решения.

2 Системы линейных уравнений над конечным полем

2.1 Равновероятный случай, теоремы инвариантности

Рассмотрим линейную систему уравнений над полем $\mathrm{GF}(q)$:

$$a_{i_1}x_1 + \ldots + a_{i_n}x_n = b_n, \quad i = 1, \ldots, t. \tag{6}$$

Пусть элементы матрицы A_t системы уравнений (6) независимы и имеют равномерное распределение на элементах поля $\mathrm{GF}(q)$:

$$\mathsf{P}\{a_{ij} = a\} = \frac{1}{q}, \quad a \in \mathrm{GF}(q), \quad i = 1, \ldots, t, \quad j = 1, \ldots, n.$$

Обозначим $p_t(r)$ вероятность того, что ранг матрицы A_t размера $t \times n$ равен r.

Точные формулы для вероятностей $p_t(t-k)$, $k \geqslant 0$, $t \geqslant n$ приведены в [4]:

$$p_t(t-k) = q^{-k(n-t+k)} \prod_{j=k+1}^{t} (1 - q^{-j}) \prod_{j=n-t+k+1}^{n} (1 - q^{-j}) \Bigg/ \prod_{j=1}^{t-k} (1 - q^{-j}). \tag{7}$$

Более удобная для вычисления этих вероятностей формула содержится в следующем утверждении.

Теорема 2 ([5]). *Пусть элементы матрицы A_t независимы и равномерно распределены в поле* $\mathrm{GF}(q)$. *Тогда для любых $k \geqslant 0$, $t \leqslant n$*

$$p_t(t-k) = q^{-k(n-t+k)} \prod_{i=1}^{k} (1 - q^{-i}) \prod_{j=n-t+k+1}^{\infty} (1 - q^{-j})\,[1 + C],$$

где $-kq^{-(t-k+1)} \leqslant C \leqslant 2q^{-n}$, а при $k = 0$ первое произведение следует положить равным единице.

Если $t > n$, то в приведенной выше формуле достаточно поменять местами t и n, так как вероятностная структура матрицы не меняется при транспонировании.

Очевидно, что для равновероятно распределённых элементов матрицы системы (6) и правых частей

$$\mathsf{E}\xi_t = q^{n-t}, \quad \mathsf{E}\eta_t = 1 + (q^n - 1)q^{-t}.$$

Поэтому, используя лемму 2, находим простую оценку

$$\mathsf{P}\{\xi_t > 0\} > [1 + q^{t-n}]^{-1}.$$

Предположим, что для распределений коэффициентов булевой системы (6) выполняются неравенства

$$0 < \delta \leqslant \mathsf{P}\{a_{ij} = 1\} \geqslant 1 - \delta, \quad i = 1, \ldots, t, \quad j = 1, \ldots, n. \tag{8}$$

Определим при каком $\delta \neq 1/2$ асимптотическое распределение числа решений однородной системы (6) будет таким, как и в равновероятном случае, т. е. при $\delta = 1/2$. Ряд вопросов, относящихся к проблеме инвариантности некоторых характеристик систем случайных линейных уравнений, решены Коваленко И. Н. и его учениками Левитской А. А. и Масолом В. И. [6, 7]. Мы приведём здесь в упрощённом виде только некоторые из этих результатов.

Среднее число решений однородной системы линейных уравнений (6) в равновероятном случае

$$\mathsf{E}\eta_t = 1 + (2^n - 1)/2^t.$$

Очевидно, что необходимым условием совпадения предельного распределения числа решений однородной системы, коэффициенты которой имеют распределение (8), с распределением (7) является совпадение в пределе их математических ожиданий. Более того, как это следует из теоремы 1.1 работы автора [5], совпадение предельных значений математических ожиданий в ряде случаев может оказаться и достаточным условием совпадения предельных распределений чисел решений.

Теорема 3. *Пусть элементы матрицы $A_t = \|a_{ij}\|$ размера $t \times n$ независимы и для вероятностей $p_{ij} = \mathsf{P}\{a_{ij} = 1\}$ справедливы неравенства*

$$\delta \leqslant p_{ij} \leqslant 1 - \delta, \quad i = 1, \ldots, t, \quad j = 1, \ldots, n$$

для $\delta = (\ln n + \omega_n)/n$, $\omega_n \to \infty$ при $n \to \infty$. Тогда среднее число нетривиальных решений однородной системы

$$\mathsf{E}(\eta_t - 1) = 2^{n-t} + o(1)$$

при $n \to \infty$, $n - t = O(1)$.

Далее было доказано, в частности, вначале для $\delta \geqslant \varepsilon > 0$, а затем для случая $p_{ij} = (\ln n + \omega_n)/n$, что предельное распределение числа решений такой однородной системы совпадает с предельным распределением в равновероятном случае.

И, наконец, в работе [8] Масолу В. И. удалось доказать для случая

$$p_{ij} = (\ln n + x)/n, \quad i = 1, \ldots, t, \quad j = 1, \ldots, n,$$

x — фиксированное число, что в пределе распределение ранга матрицы системы после выбрасывания пустых строк и столбцов такое же, как и в равновероятном случае (7).

2.2 Случай разреженной матрицы системы

Пусть элементы матрицы A_t независимы и имеют следующее распределение на элементах поля $\mathrm{GF}(q)$:

$$\mathsf{P}\{a_{ij} = 0\} = 1 - \Delta, \quad \mathsf{P}\{a_{ij} = a \neq 0\} = \frac{\Delta}{q-1},$$
$$a \in \mathrm{GF}(q), \quad i = 1, \ldots, t, \quad j = 1, \ldots, n,$$

где $\Delta = (\ln n + x)/n$, x — фиксированное число. Для таких случайных матриц A_t характерно то, что в них с положительной вероятностью появляются нулевые строки и столбцы.

Введём несколько определений и обозначений. Ранг матрицы A над полем $\mathrm{GF}(q)$ назовём детерминантным рангом (d.r. A) и примем обозначение $d_k = \mathsf{P}\{\mathrm{d.r.}\, A = k\}$. Перманентным рангом матрицы A (p.r. A) назовём максимально возможное число её ненулевых элементов, находящихся в разных строках и столбцах, и введём обозначение $p_k = \mathsf{P}\{\mathrm{p.r.}\, A = k\}$. Рангом линий или линейным рангом матрицы A (l.r. A) назовём максимальный порядок минора, все строки и столбцы которого ненулевые. Очевидно,

что ранг линий равен $\min(t-\mu, n-\nu)$, где μ — число нулевых строк, а ν — число нулевых столбцов матрицы A. Пусть $l_k = \mathsf{P}\{\text{l.r.}\, A = k\}$.

В работе [9] П. Эрдёш и А. Реньи впервые обратили внимание на то, что в некоторых случаях

$$\lim_{n\to\infty} p_m = \lim_{n\to\infty} l_m, \quad m = \min(t, n).$$

Затем в работе [10] автор показал, что в этом случае

$$\lim_{n\to\infty} p_{m-k} = \lim_{n\to\infty} l_{m-k}$$

для любого $k \geqslant 0$.

Исследование поведения введённых трёх рангов привело к следующему результату [11].

Теорема 4. *Пусть* $|t-n| \to \infty$ *при* $n \to \infty$, $\Delta = (\ln n + x)/n$, $x = O(1)$. *Тогда*

1) *при* $n - t \to \infty$

$$\mathsf{P}\{\text{d.r.}\, A = \text{p.r.}\, A = \text{l.r.}\, A = t - \mu\} \to 1,$$
$$\mathsf{P}\{\text{d.r.}\, A = t - k\} - \frac{(\alpha e^{-x})^k}{k!} \exp\{-\alpha e^{-x}\} \to 0, \quad \alpha = t/n;$$

2) *при* $t - n \to \infty$

$$\mathsf{P}\{\text{d.r.}\, A = \text{p.r.}\, A = \text{l.r.}\, A = n - \nu\} \to 1,$$
$$\mathsf{P}\{\text{d.r.}\, A = n - k\} - \frac{(e^{-x-\beta})^k}{k!} \exp\{-e^{-x-\beta}\} \to 0,$$

где $\beta = ((t-n)/n)\ln n$.

2.3 Матрицы с фиксированным числом ненулевых элементов в каждой строке

Пусть в каждой строке матрицы A находится ровно одна единица. Этот случай интересен тем, что здесь совпадают все три введённых в п. 2.2 ранга матрицы A и задача нахождения распределения ранга матрицы A сводится к известной задаче о распределении числа пустых ящиков в классической задаче о дробинках.

Если в каждой строке матрицы A находятся две единицы, то мы приходим к системе булевых уравнений

$$x_{i_k} + x_{j_k} = a_k, \quad k = 1, \ldots, t,$$

которой соответствует случайный граф с n вершинами $x_1, \ldots, x_n$ и t рёбрами $(i_1, j_1), \ldots, (i_t, j_t)$ соответственно с весами $a_1, \ldots, a_t$. Такие случайные графы изучались Степановым В. Е. ([12], [13]) и Колчиным В. Ф. [14]. Приведём один из результатов Степанова В. Е.

Пусть $t/n \to \alpha < 1/2$ при $n \to \infty$. Тогда для любого фиксированного $k \geqslant 0$

$$\lim_{n\to\infty} \mathsf{P}\{\text{d.r.}\, A = t - k\} = \frac{\lambda^k}{k!} e^{-\lambda},$$

где $\lambda = (1/2)\ln(1/(1-2\alpha))$. Если же в каждой строке матрицы A находится ровно две единицы, т. е. в соответствующем графе нет петель, то в этом утверждении λ заменяется на $\lambda_1 = -\alpha + (1/2)\ln(1/(1-2\alpha))$.

В работах [15], [16], [17] изучались системы случайных линейных уравнений с $r \geqslant 3$ неизвестными в каждом уравнении. Каждая из C_n^r возможных расстановок r единиц в строке имеет одну и ту же вероятность $(C_n^r)^{-1}$ быть выбранной. Изучено асимптотическое поведение различных характеристик таких случайных систем уравнений. В частности, показано, что для некоторого $a_r < 1$, $r \geqslant 3$ при $n \to \infty$ $\mathsf{E}\xi_t/q^{n-t} \to \infty$, если $\lim_{n\to\infty} t/n < a_r$, и $\mathsf{E}\xi_t/q^{n-t} \to 0$ для $\lim_{n\to\infty} t/n > a_r$.

3 Системы нелинейных уравнений

3.1 Геометрический метод исследования

Рассмотрим нелинейную систему уравнений над полем

$$f_i(x_1, \ldots, x_n) = b_i, \quad i = 1, \ldots, t. \tag{9}$$

Пусть ξ — число решений этой системы, а X_i — множество решений i-го уравнения. Тогда

$$\mathsf{P}\{\xi = k\} = \mathsf{P}\left\{ \left| \bigcap_{j=1}^{t} X_j \right| = k \right\}.$$

Занумеруем все наборы значений неизвестных $\overline{x} = (x_1, \ldots, x_n)$ системы (9) индексами $1, 2, \ldots, q^n$ и построим матрицу решений $A = \|a_{ij}\|$ размера $t \times q^n$ по следующему правилу:

$$a_{ij} = \begin{cases} 1, & \text{если } x_j \in X_i, \\ 0, & \text{если } x_j \notin X_i. \end{cases}$$

Очевидно, что число решений системы (9) равно числу единичных столбцов матрицы решений A.

Функция f_i на множестве всех векторов $\overline{x}$ задаёт разбиение

$$X = \bigcup_b X_{i,b}, \quad X_{i,b} = \{\overline{x} : f_i(\overline{x}) = b\}.$$

Выбор правой части b_i указывает множество решений i-го уравнения $X_i = X_{i,b_i}$. Вероятностная мера, заданная на множестве пар (f_i, b_i), индуцирует вероятностную меру на подмножествах X_i множества X:

$$\mathsf{P}(X_i) = \sum_{(f_i, b_i) : X_{i,b_i} = X_i} p(f_i, b_i).$$

Поэтому можно перенести внимание на изучение случайных $(0,1)$-матриц A и, что то же самое, случайных подмножеств множества X.

Примеры использования такого подхода к изучению нелинейных систем уравнений приведены в работах [18], [19], [20], [5].

3.2 Комбинаторный метод исследования

Рассмотрим систему уравнений

$$f_i(x_{s_{i1}}, \ldots, x_{s_{ir}}) = a_i, \quad i = 1, \ldots, t \tag{10}$$

Пусть:

- $s_i = (s_{i1}, \ldots, s_{ir})$ — равновероятная выборка без возвращения из $\{1, \ldots, n\}$;
- $(s_1, \ldots, s_t)$ независимы в совокупности;
- $(f_1, \ldots, f_t)$ — независимые одинаково распределённые случайны величины, принимающие значения из некоторого класса функций Φ;
- $(a_1, \ldots, a_t)$ — некоторые случайные величины.

Матрицу из нулей и единиц $B = \|b_{i,j}\|$ с элементами

$$b_{ij} = \begin{cases} 1, & \text{если } j \in \{s_{i1}, \ldots, s_{ir}\}, \\ 0, & \text{если } j \notin \{s_{i1}, \ldots, s_{ir}\}, \end{cases}$$

назовём матрицей инциденций неизвестных и уравнений системы (10). Для заведомо совместных систем уравнений число решений η не меньше q^ν, где ν — число отсутствующих неизвестных в уравнениях системы или число пустых (нулевых) столбцов матрицы B. Если $\mathsf{E}\eta - \mathsf{E}q^\nu = o(1)$ при $n \to \infty$, то $\eta \to q^\nu$ по вероятности при $n \to \infty$, поскольку

$$\mathsf{P}\{\eta \neq q^\nu\} = \mathsf{P}\{\eta > q^\nu\} \leqslant \mathsf{E}\eta - \mathsf{E}q^\nu = o(1).$$

В этом случае система (10) имеет единственное решение относительно тех неизвестных, которые присутствуют в системе.

Рассмотрим строение матрицы инциденций B системы (10). Матрица B принадлежит множеству матриц размера $t \times n$ с r единицами и $n - r$ нулями в каждой строке. Число таких матриц $[C_n^r]^t$ и на этом множестве задана мера: $\mathsf{P}(B) = [C_n^r]^{-t}$ Скажем, что осуществилось событие B_k, если матрица B перестановкой столбцов приводится к лестничному виду с k ступеньками длины $r-1$ и $t-k$ ступеньками длины r. При этом каждая ступенька может иметь пересечение (общий столбец с единицами) не более чем с одной ступенькой.

Строение матриц в самом начале процесса сокращения длин ступенек описывается следующим утверждением [5].

Теорема 5. *Пусть* $r = o\left(n^{1/2}\right)$, $t = \sqrt{2\lambda n}/r$, $\lambda = O(1)$. *Тогда*

$$\mathsf{P}\{B_k\} = \frac{\lambda^k}{k!} e^{-lambda} = o(1).$$

Рассмотрим возможные применения этого утверждения. Пусть X_i — множество решений i-го уравнения системы (10), A_k — множество индексов неизвестных, входящих в k-е уравнение,

$$\widetilde{p}_s = \mathsf{P}\left\{\, |X_i \cap X_j| = sq^{n-2r+1} \;\middle|\; |A_i \cap A_j| = 1\right\},$$
$$q_s = \mathsf{P}\{|X_i| = sq^{n-r}\}, \quad s \geqslant 0.$$

Если осуществилось событие B_k, то система имеет

$$\xi = q^{n-tr+k} \prod_{i=1}^{k} \eta_i \prod_{j=1}^{t-2k} \xi_j$$

решений, где все случайные величины независимы в совокупности и

$$\mathsf{P}\{\eta_i = j\} = \widetilde{p}_j, \quad \mathsf{P}\{\xi_i = j\} = q_j, \quad i = 1, \ldots, t, \quad j \geqslant 0.$$

Зная $\mathsf{P}\{B_k\}$, $k \geqslant 0$, нетрудно найти распределение числа решений ξ. В частности,

$$\begin{aligned}\mathsf{P}\{\xi > 0\} &= \sum_{k \geqslant 0} \mathsf{P}\{B_k\}(1-\widetilde{p}_0)^k (1-q_0)^{t-2k} + o(1) \\ &= (1-q_0)^t \exp\left\{-\lambda \frac{\widetilde{p}_0 - 2q_0 + q_0^2}{(1-q_0)^2}\right\} + o(1).\end{aligned}$$

3.3 Комбинаторно-алгебраические методы изучения систем уравнений

Будем считать, что неизвестное x_i несущественно входит в систему булевых уравнений относительно решения $\overline{x}^0$ этой системы, если указанной системе уравнений удовлетворяет вектор $\overline{x}^0 \oplus e_i$, отличающийся от вектора $\overline{x}^0$ только i-й координатой.

Заметим, что отсутствующие в системе уравнений неизвестные являются несущественными для любого вектора из множества решений этой системы. Для однородной линейной системы только отсутствующие неизвестные являются несущественными относительно решения $\overline{0}$. Поэтому введённое понятие имеет смысл рассматривать только для нелинейных уравнений.

Определение несущественности неизвестного несколько в ином виде присутствует в работе Копытцева В. А. [21], в которой приводится довольно общий результат о связи числа решений нелинейной системы с числом несущественных неизвестных. Неявно это понятие используется и в других работах (см., например, [22]).

В качестве простейшего примера использования алгебраического подхода к анализу систем нелинейных уравнений рассмотрим заведомо совместную систему булевых уравнений (10) с r неизвестными в каждом уравнении. Полагаем, что функции $f_1, \ldots, f_t$ выбираются независимо и равновероятно из множества всех булевых функций от r переменных.

Пусть ν_i есть индикатор события

$$\left\{ f_i\left(\overline{x}^0 \oplus e_i\right) = f_i\left(\overline{x}^0\right), \quad i = 1, \ldots, t \right\},$$

а случайная величина

$$\nu = \nu_1 + \ldots + \nu_n$$

есть число неизвестных, несущественных для системы уравнений (10) относительно решения $\overline{x}^0$.

В работе [23] доказаны следующие утверждения.

Теорема 6. *Пусть*

$$t = \frac{2n(\ln n + z)}{r}, \quad z = O(1), \quad \overline{\lim} \frac{r}{\ln n} < 1.$$

Тогда при $n \to \infty$ *для любого фиксированного* $k = 0, 1, 2, \ldots$

$$\mathsf{P}\{\nu = k\} = \frac{(e^{-z})^k}{k!} \exp\{-e^{-z}\} + o(1).$$

Теорема 7. *В условиях теоремы 6 число решений* η_t *заведомо совместной системы уравнений* (10) *стремится по вероятности при* $n \to \infty$ *к случайной величине* 2^ν, *где* ν — *число несущественных неизвестных относительно решения* $\overline{x}^0$, *и, следовательно, при любом фиксированном* $k = 0, 1, 2, \ldots$

$$\mathsf{P}\left\{\eta_t = 2^k\right\} = \frac{(e^{-z})^k}{k!} \exp\{-e^{-z}\} + o(1).$$

Заметим, что в рассматриваемом случае с вероятностью, стремящейся к 1 при $n \to \infty$, все неизвестные существенно присутствуют в системе, но некоторые из них несущественны относительно истинного решения $\overline{x}^0$.

К алгебраическим методам анализа относятся и те методы, которые выделяют особые свойства специальных классов систем уравнений, например, «запреты» (см. [24]).

4 Системы уравнений с искажённой правой частью

4.1 Метод максимума правдоподобия

Рассмотрим заведомо совместную систему булевых уравнений

$$f_i(x_1, \ldots, x_n) = a_i, \quad i = 1, \ldots, t. \tag{11}$$

Предположим, что правая часть в (11) искажена, т. е. наблюдаются значения

$$b_i = a_i \oplus \varepsilon_i, \quad i = 1, \ldots, t,$$

где $\varepsilon_1, \ldots, \varepsilon_t$ — не известные нам реализации независимых случайных величин $\xi_1^0, \ldots, \xi_t^0$, $\mathsf{P}\{\xi_i^0 = 1\} = p < 1/2$. Система уравнений вида

$$f_i(x_1, \ldots, x_n) = b_i, \quad i = 1, \ldots, t \tag{12}$$

называется системой уравнений с искажённой правой частью. Заметим, что при $p = 1/2$ система (12) является случайной, а при $p = 0$ — заведомо совместной. Таким образом, система (12) занимает промежуточное положение между случайной и заведомо совместной случайной системами, так как для любого $i \geqslant 1$ выполняются неравенства

$$\frac{1}{2} < \mathsf{P}\left\{b_i = f_i\left(x_1^0, \ldots, x_n^0\right)\right\} = 1 - p < 1.$$

В криптографии такие системы уравнений возникают при замене сложных нелинейных функций на более простые статистические аналоги [25], а в теории кодирования такие системы уравнений появляются более естественным образом [26].

Каждому вектору $\overline{x}$ соответствует вектор ошибок

$$\overline{\varepsilon}(\overline{x}) = \overline{f}(\overline{x}) \oplus \overline{b}.$$

Если вектор $\overline{x}^0$ имеет равномерное распределение на множестве возможных решений, то вектор $\overline{x}^*$, удовлетворяющий условию

$$\max_{\overline{x}} \mathsf{P}\left\{\overline{\xi}^0 = \overline{\varepsilon}(\overline{x})\right\} = \mathsf{P}\left\{\overline{\xi}^0 = \overline{\varepsilon}(\overline{x}^*)\right\},$$

естественно считать максимально правдоподобным.

Такой способ построения оценки вектора $\overline{x}^0$ называется методом максимального правдоподобия. В теории кодирования аналогичная процедура для декодирования блоковых кодов, задаваемых преобразованием

$$(x_1, \ldots, x_n) \to (f_1(x_1, \ldots, x_n), \ldots, f_t(x_1, \ldots, x_n)),$$

называется методом декодирования в ближайшее кодовое слово. Основной задачей здесь является оценка величины $\mathsf{P}\left\{\overline{x}^* = \overline{x}^0\right\}$, которая характеризует надёжность метода максимума правдоподобия. Приведём оценки этой величины для случаев случайной и фиксированной левых частей системы.

Теорема 8. *Пусть выполняются следующие условия:*

1) *функции $f_1, \ldots, f_t$ независимо выбираются из некоторого множества функций Φ;*

2) $\mathsf{P}\left\{f_i(\overline{x}) \neq f_i\left(\overline{x}^0\right)\right\} \geqslant \overline{p}$ *для любого* $\overline{x} \neq \overline{x}^0$, $i = 1, \ldots, t$; $0 < \overline{p} \leqslant 1$;

3) *при $n \to \infty$ мощность множества возможных решений $|X| \to \infty$,*

$$t = (1 + \theta_n) \ln |X| \Big/ \ln \left[\left(\frac{p}{p^*}\right)^p \left(\frac{1-p}{1-p^*}\right)^{1-p}\right],$$

$$p^* = p + \overline{p}(1-2p), \quad \lim_{n\to\infty} \theta_n > 0.$$

Тогда построенная методом максимума правдоподобия оценка решения $\overline{x}^0$ состоятельна:

$$\mathsf{P}\left\{\overline{x}^* = \overline{x}^0\right\} \to 1, \quad n \to \infty.$$

Следствие. *Если в условиях теоремы 8 $\overline{p} = 1/2$, $|X| = 2^n$, то при*

$$t = (1 + \theta_n) n \Big/ \log_2 \left\lfloor 2p^p(1-p)^{1-p} \right\rfloor, \quad \lim_{n\to\infty} \theta_n > 0$$

оценка $\overline{x}^*$ истинного решения $\overline{x}^0$ состоятельна.

Пусть $f_1, \ldots, f_t$ — известные фиксированные функции. Разобьём множество всех n-мерных двоичных векторов $\overline{x}$ на классы

$$X_k = \left\{\overline{x} : \sum_{i=1}^{t} \left[f_i(\overline{x}) \oplus f_i\left(\overline{x}^0\right)\right] = k\right\}, \quad \overline{x}^0 \in X_0, \quad k = 0, \ldots, t.$$

Набор чисел $(|X_0|, \ldots, |X_t|)$ естественно назвать спектром расстояний системы (12) относительно решения $\overline{x}^0$ (по аналогии со спектром расстояний кода, задаваемого соответствующим преобразованием $\overline{f}$).

Теорема 9. *Пусть левая часть системы* (12) *фиксирована,* $(|X_0|, \ldots, |X_t|)$ *— спектр расстояний системы* (12) *относительно истинного решения $\overline{x}^0$. Тогда вероятность того, что метод максимума правдоподобия даёт правильное решение системы* (12)*, удовлетворяет неравенству*

$$\mathsf{P}\left\{\overline{x}^* = \overline{x}^0\right\} \geqslant 1 - \sum_{k \geqslant 1} |X_k| p(k),$$

где

$$p(k) = \sum_{i \geqslant k/2} C_k^i p^i (1-p)^{k-i} \leqslant \sigma^k, \quad \sigma = \sqrt{4p(1-p)}.$$

4.2 Критерии, выделяющие систему уравнений с искажённой правой частью из множества случайных систем уравнений

Для криптографических приложений необходимо построить критерий, различающий гипотезы:

- H_0 — система (12) является системой с искажённой правой частью ($p < 1/2$);
- H_1 — система (12) случайна ($p = 1/2$).

Если имеется M систем уравнений (12) и заранее известно, что только одна из них является системой уравнений с искажённой правой частью, а остальные $M - 1$ систем являются случайными, то необходимо построить быстрый критерий отбраковки случайных систем уравнений. Такая ситуация возникает, например, при переборе значений s неизвестных. В этом случае $M = 2^s$.

Наиболее просто такие критерии строятся для системы из линейных уравнений, которую можно представить как заведомо совместную систему

$$L_i(x_1, \ldots, x_n) = b_i \oplus \varepsilon_i, \quad i = 1, \ldots, t, \tag{13}$$

имеющую 2^n решений $(\overline{x}, \overline{\varepsilon}(\overline{x}))$. При $t > n$ можно исключить все неизвестные и составить уравнения относительно ошибок $\varepsilon_1, \ldots, \varepsilon_t$. Из этих уравнений можно составить некоторые s-членные уравнения для ошибок. Пусть $m_s(j)$ — число s-членных уравнений, которые содержат ε_j и не содержат другие ошибки более одного раза, а $m_{s1}(j)$ — число единиц в правых частях этих $m_s(j)$ уравнениях, $s \geqslant 1$. Положим

$$s_j = m_1(j) + m_2(j) + \ldots + m_t(j),$$

H_{0a} — гипотеза H_0 при $\varepsilon_j^0 = a$. Справедливо следующее утверждение [27].

Теорема 10. *Статистика*

$$Q(j) = \sum_{s \geqslant 1} \Delta^s [m_s(j) - 2m_{s1}(j)]$$

разделяет гипотезы $H_{00}(j)$ ($\varepsilon_j^0 = 0$), $H_{01}(j)$ ($\varepsilon_j^0 = 1$), $H_1(j)$ с вероятностью, стремящейся к единице при $s_j \to \infty$, если при $s_j \to \infty$

$$\sum_{s \geqslant 1} \Delta^{2s-2} m_s(j) \to \infty.$$

Рассмотрим ещё один простой пример. Пусть система (13) имеет вид

$$x_i \oplus x_j = a_{ij} \oplus \varepsilon_{ij}, \quad 1 \leqslant i < j \leqslant n. \tag{14}$$

В системе $n(n-1)/2$ уравнений и два истинных решения $\overline{x}^0$ и $\overline{x}^0 \oplus \overline{1}$. Полагаем $x_n = 0$ и находим $x^i = a_{in} \oplus \varepsilon_{in}$, $i = 1, \ldots, n-1$. Подставим эти выражения в исходную систему (14), получим систему из трёхчленных соотношений для ошибок

$$\varepsilon_{ij} \oplus \varepsilon_{in} \oplus \varepsilon_{jn} = a_{ij} \oplus a_{in} \oplus a_{jn} = c_{ijn}, \quad 1 \leqslant i < j \leqslant n-1.$$

Оказывается верным следующее утверждение [28].

Теорема 11. *Пусть $\Delta^2 \sqrt{n} \to \infty$ при $n \to \infty$. Если гипотеза H_0 принимается при*

$$S_1 = \left| \sum_{j=2}^{n-1} c_{1jn} - \frac{n-2}{2} \right| > \frac{\Delta^2 (n-2)}{4},$$

а гипотеза H_1 — при $S_1 \leqslant \Delta^2(n-2)/4$, то в пределе при $n \to \infty$ этот критерий разделяет гипотезы H_0 и H_1 с вероятностями ошибок первого и второго рода, стремящимися к нулю. При этом при справедливости гипотезы H_0 правильные значения $x_1^0, \ldots, x_k^0$ определяются с вероятностью, стремящейся к единице, для любого фиксированного $k \geqslant 1$.

С некоторыми методами решения систем уравнений с искажёнными правыми частями можно ознакомиться в работах [2], [29], [30], [31], [32].

Литература

[1] Балакин Г. В. Введение в теорию случайных систем уравнений. В сб.: Труды по дискретной математике. Т. 1. М.: ТВП, 1997, с. 1–18.

[2] Балакин Г. В., Никольский Ю. Б. Последовательное применение метода максимума правдоподобия к решению систем уравнений с мешающими параметрами. Обозрение прикл. промышл. матем., 1995, т. 2, вып. 3, с. 468–476.

[3] Сачков В. Н. Случайные неравновероятные покрытия и функциональные уравнения. В сб.: Труды по дискретной математике. Т. 5. М.: ФИЗМАТЛИТ, 2002, с. 205–218.

[4] Landsberg G. Über eine Anzahlbestimmung und eine damit zusammenhangende Reihe. J. Reine Angew. Math., III, 1895, p. 87–88.

[5] Балакин Г. В. Системы случайных уравнений над конечным полем. В сб.: Труды по дискретной математике. Т. 2. М.: ТВП, 1998, с. 21–37.

[6] Коваленко И. Н., Левитская А. А., Савчук М. Н. Избранные задачи вероятностной комбинаторики. Киев: Наукова думка, 1986, 224 с.

[7] Kovalenko I. N., Levitskaya A. A. Stochastic properties of systems of random linear equations over finite algebraic structures. В кн.: Вероятностные методы дискретной математики. Труды третьей Петрозаводской конференции. (Петрозаводск, 12–15 мая 1992 г.). Москва/Utrecht: ТВП/VSP, 1993, с. 64–70.

[8] Масол В. И. Расширение области инвариантности для случайных булевых матриц. Кибернетика, 1980, № 3, с. 125–128.

[9] Erdős P., Renyi A. On random matrices. Magyar Tud. Akad. Mat. Kutato Ind. Közl., 1963, **9**, 3, 455–461.

[10] Балакин Г. В. О случайных матрицах.Теория вероятн. и её примен., 1967, т. 12, вып. 2, с. 346–353.

[11] Балакин Г. В. Распределение ранга случайных матриц над конечным полем. Теория вероятн. и её примен., 1968, т. 13, вып. 4, с. 631–641.

[12] Степанов В. Е. Фазовые переходы в случайных графах. Теория вероятн. и её примен., 1970, т. 15, вып. 2, с. 200–216.

[13] Степанов В. Е. О некоторых особенностях строения случайного графа вблизи критической точки. Теория вероятн. и её примен., 1987, т. 32, вып. 4, с. 633–657.

[14] Колчин В. Ф. Системы случайных уравнений. М.: МИЭМ, 1988, 80 с.

[15] Балакин Г. В., Колчин В. Ф., Хохлов В. И. Гиперциклы в случайном гиперграфе. Дискретн. матем., 1991, т. 3, № 3, с. 102–108.

[16] Колчин В. Ф. О пороговом эффекте для систем случайных уравнений. В сб.: Труды по дискретной математике. Т. 2. М.: ТВП, 1998, с. 183–190.

[17] Шаповалов А. В. Вероятность совместности случайных систем булевых уравнений. Дискретн. матем., 1995, т. 7, вып. 2, с. 146–159.

[18] Balakin G. V. On the number of solutions of systems of pseudo-boolean random equations. В кн.: Вероятностные методы дискретной математики. Труды третьей Петрозаводской конференции. (Петрозаводск, 12–15 мая 1992 г.). Москва/Utrecht: ТВП/VSP, 1993, с. 71–98.

[19] Михайлов В. Г. Предельные теоремы для случайного покрытия конечного множества и числа решений системы случайных уравнений. Теория вероятн. и её примен., 1996, т. 41, вып. 2.

[20] Сачков В. Н. Развитие комбинаторно-вероятностных направлений в математике и криптографии. В кн.: Московский университет и развитие криптографии в России. Материалы конференции в МГУ 17–18 октября 2002 г. М.: МЦНМО, 2003, с. 33–48.

[21] Копытцев В. А. О распределении числа решений случайных заведомо совместных систем уравнений. Теория вероятн. и её примен., 1995, т. 40, вып. 2, с. 430–437.

[22] Балакин Г. В. Графы систем двучленных уравнений с булевыми неизвестными. Теория вероятн. и её примен., 1995, т. 40, вып. 2, с. 241–259.

[23] Балакин Г. В. Системы случайных булевых уравнений со случайным выбором неизвестных в каждом уравнении. В сб.: Труды по дискретной математике. Т. 3. М.: ФИЗМАТЛИТ, 2000, с. 21–28.

[24] Сумароков С. Н. Запреты двоичных функций и обратимость для одного класса кодирующих устройств. Обозрение прикл. промышл. матем., сер. дискретн. матем., 1994, т. 1, вып. 1, с. 33–55.

[25] Логачёв О. А., Сальников А. А., Ященко В. В. Криптографические свойства дискретных функций. В кн.: Московский университет и развитие криптографии в России. Материалы конференции в МГУ 17–18 октября 2002 г. М.: МЦНМО, 2003, с. 174–199.

[26] Сидельников В. М. Криптография и теория кодирования. В кн.: Московский университет и развитие криптографии в России. Материалы конференции в МГУ 17–18 октября 2002 г. М.: МЦНМО, 2003, с. 49–84.

[27] Балакин Г. В. Критерии, выделяющие заведомо совместную систему уравнений с искажённой правой частью. - В сб.: Труды по дискретной математике. Т. 4. М.: ФИЗМАТЛИТ, 2001, с. 7–16.

[28] Балакин Г. В. Последовательный критерий выделения системы линейных уравнений с искажённой правой частью. В сб.: Труды по дискретной математике. Т. 5. М.: ФИЗМАТЛИТ, 2002, с. 21–28.

[29] Балакин Г. В. О возможности решения систем линейных целочисленных уравнений методом выделения и оценки отдельных неизвестных. Дискрет. матем., 1994, т. 6, вып. 1. с. 116–126.

[30] Балакин Г. В. О вероятностном подходе к решению систем уравнений с целочисленными неизвестными. Дискрет. матем., 1995, т. 7, вып. 1, с. 88–98.

[31] Колчин В. Ф. Одна задача классификации при наличии ошибок в измерениях. Дискрет. матем., 1994, т. 5, вып. 1, с. 53–66.

[32] Колчин В. Ф. Структура решений и восстановление истинного решения системы уравнений с искажёнными правыми частями. В сб.: Труды по дискретной математике. Т. 5. М.: ФИЗМАТЛИТ, 2002, с. 93–102.

Об определении основных криптографических понятий

Б. А. Погорелов, А. В. Черемушкин, С. И. Чечета

Аннотация

В докладе предлагается вариант определения основных криптографических понятий, основанный на введении обобщающего понятия криптосистемы. Определяются виды криптографических систем, основными из которых являются системы шифрования, идентификации, имитозащиты, цифровой подписи, и ключевая система, обеспечивающая работу остальных систем.

За последние годы появилось большое число публикаций и в том числе глоссариев по криптографии на русском языке, причем их число постоянно растет. В них используются в основном различные переводы англоязычных терминов. Неточный перевод или неправильное их употребление зачастую приводят к засорению используемой терминологии и создают трудности для понимания. Не улучшает ситуацию и большое число изданных словарей терминов по безопасности, которые также придерживаются различных подходов. В качестве удачных можно привести примеры переводов иностранных статей [1] и [2], которые оказали в свое время большое воздействие на отечественную криптографическую терминологию.

Академия криптографии РФ начала работу по систематизации криптографической терминологии, причем в этом году к работе подключился Московский государственный университет. Необходимость выработки общепринятого понимания основных понятий в данной области назрела давно, однако решение этой задачи связано с большими трудностями в силу необходимости учета многочисленных нюансов, связанных с использованием сходной терминологии для многообразия конкретных областей применения криптографии, каждая из которых обладает своей спецификой. Этому способствовали закрытость многих работ, а также существенное расширение в последние годы предмета криптографии.

В докладе предлагается вариант определения некоторых основных понятий. Он построен по следующему принципу: вводится обобщающее понятие криптосистемы и выделяется четыре базовых вида криптографической системы — шифрования, идентификации, имитозащиты и цифровой подписи, а также ключевая система, обеспечивающая работу этих систем.

Каждый вид криптосистемы определяется путем введения еще одного-двух уровня понятий. Таким образом, выстраиваются 3–4 верхних уровня дерева основных понятий, представляющих собою фрагмент остовного дерева в графе зависимости между различными криптографическими терминами. Предлагаемый список понятий обладает определенной внутренней замкнутостью, поскольку все определения криптографических понятий в основном используют термины из данного списка (это выделено курсивом), и легко дополняется введением следующих уровней.

Он уточняет и продолжает опубликованный в работе [3] список терминов и в значительной мере опирается на книги[4, 5, 6, 7] и др.

Основные термины

1 Криптография

Криптография — до 70-х гг. XX в. — область науки и практической деятельности, связанная с разработкой, применением и анализом *шифрсистем*; в настоящее время — область науки, техники и практической деятельности, связанная с разработкой, применением и анализом *криптографических систем* защиты информации. Основными функциями криптографических систем являются обеспечение *конфиденциальности* и *аутентификации* различных аспектов информационного взаимодействия.

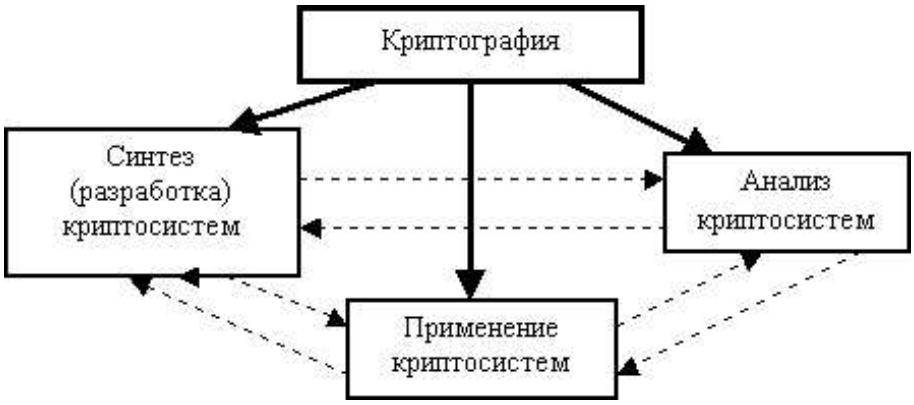


Рис. 1:

Источником угроз при решении криптографических задач считаются преднамеренные действия противника или недобросовестного участника информационного взаимодействия, а не случайные искажения информации вследствие помех, отказов и т. п.

Конфиденциальность — защищенность информации от ознакомления с ее содержанием со стороны лиц, не имеющих права доступа к ней.

Аутентификация — установление (то есть проверка и подтверждение) подлинности различных аспектов информационного взаимодействия: сеанса связи, *сторон* (идентификация), *содержания* (имитозащита) и *источника* (установление авторства) передаваемых сообщений, времени взаимодействия и т. д. Является важной составной частью проблемы обеспечения достоверности получаемой информации. Особенно остро эта проблема стоит в случае не доверяющих друг другу сторон, когда источником угроз может служить не только третья сторона (противник), но и сторона, с которой осуществляется информационное взаимодействие.

Диаграмма на рис. 1 иллюстрирует определение криптографии и показывает основные составляющие ее части. Пунктирные стрелки показывают тесные взаимосвязи между этими тремя составляющими.

2 Виды криптосистем

Система криптографическая (криптосистема) — система обеспечения безопасности защищенной сети, использующая *криптографические средства*. В качестве подсистем может включать системы *шифрования*, *идентификации*, *имитозащиты*, *цифровой подписи* и др., а также ключевую систему, обеспечивающую работу остальных систем. В основе выбора и построения криптосистемы лежит условие обеспечения *криптографической стойкости*. В зависимости от ключевой системы различают *симметричные* и *асимметричные* криптосистемы.

Средства криптографические — в широком смысле — методы и средства обеспечения безопасности информации, использующие *криптографические преобразования информации*; в узком смысле — средства, реализованные в виде документов, механических, электро-механических, электронных технических устройств или программ, предназначенные для выполнения функций *криптографической системы*.

Криптографическое преобразование информации — преобразование информации с использованием одного из *криптографических алгоритмов*, определяемое целевым назначением *криптографической системы*.

Симметричные криптосистемы — криптосистемы с симметричными (секретными) ключами. Симметричность означает здесь, что ключи, задающие пару взаимно обратных криптографических преобразований, могут быть получены один из другого с небольшой трудоемкостью. Стойкость симметричной криптосистемы определяется трудоемкостью, с которой противник может вычислить любой из секретных ключей, и оценивается при общепринятом допущении, что противнику известны все элементы криптосистемы, за исключением секретного ключа.

Асимметричные криптосистемы — криптосистемы с асимметричными (секретными и откры-

Рис. 2:

тыми) ключами. Асимметричность означает здесь, что из двух ключей, задающих пару взаимно обратных криптографических преобразований, один является секретным, а другой открытым. Открытые ключи известны всем участникам защищенной сети и противнику, но каждый участник сети хранит в тайне собственный секретный ключ. Стойкость асимметричной криптосистемы определяется трудоемкостью, с которой противник может вычислить секретный ключ, исходя из знания открытого ключа и другой дополнительной информации о криптосистеме.

Шифрсистема — криптографическая система обеспечения конфиденциальности, предназначенная для защиты информации от ознакомления с ее содержанием лиц, не имеющих права доступа к ней, путем шифрования информации. Математическая модель шифрсистемы включает способ кодирования исходной и выходной информации, *шифр* и *ключевую систему.*

Система имитозащиты (обеспечения целостности) информации — криптографическая система, выполняющая функцию *аутентификации содержания* сообщения или документа и предназначенная для защиты от несанкционированного изменения информации или навязывания ложной информации. Математическая модель системы имитозащиты включает криптографический *алгоритм имитозащищенного кодирования информации* (это может быть *алгоритм шифрования*, *код аутентификации*, либо другое преобразование) и алгоритм принятия решения об истинности полученной информации, а также *ключевую систему.*

Система идентификации — криптографическая система, выполняющая функцию *аутентификации сторон* в процессе информационного взаимодействия. Математическая модель системы идентификации включает *протокол идентификации* и *ключевую систему.*

Система цифровой подписи — криптографическая система, выполняющая функцию *аутентификации источника* сообщения или документа и предназначенная для защиты от отказа субъектов от некоторых из ранее совершенных ими действий. Например, отправитель может отказаться от факта передачи сообщения, утверждая, что его создал сам получатель, а получатель легко может модифицировать, подменить или создать новое сообщение, а затем утверждать, что оно получено от отправителя. Математическая модель системы цифровой подписи включает *схему цифровой подписи* и *ключевую систему.*

Система ключевая — определяет порядок использования *криптографической системы* и включает *системы установки* и *управления ключами.*

Система установки ключей — определяет алгоритмы и процедуры генерации, распределения, передачи и проверки ключей.

Система управления ключами — определяет порядок использования, смены, хранения и архивирования, резервного копирования и восстановления, замены или изъятия из обращения скомпрометированных, а также уничтожения старых ключей. Целью управления ключами является нейтрализация таких угроз, как: компрометация конфиденциальности секретных ключей, компрометация аутентичности секретных или открытых ключей, несанкционированное использование секретных или открытых ключей, например использование ключа, срок действия которого истек.

Система ключевая симметричной криптосистемы — основана на использовании симметричных (секретных) ключей. Основными проблемами таких систем являются построение *системы установки*

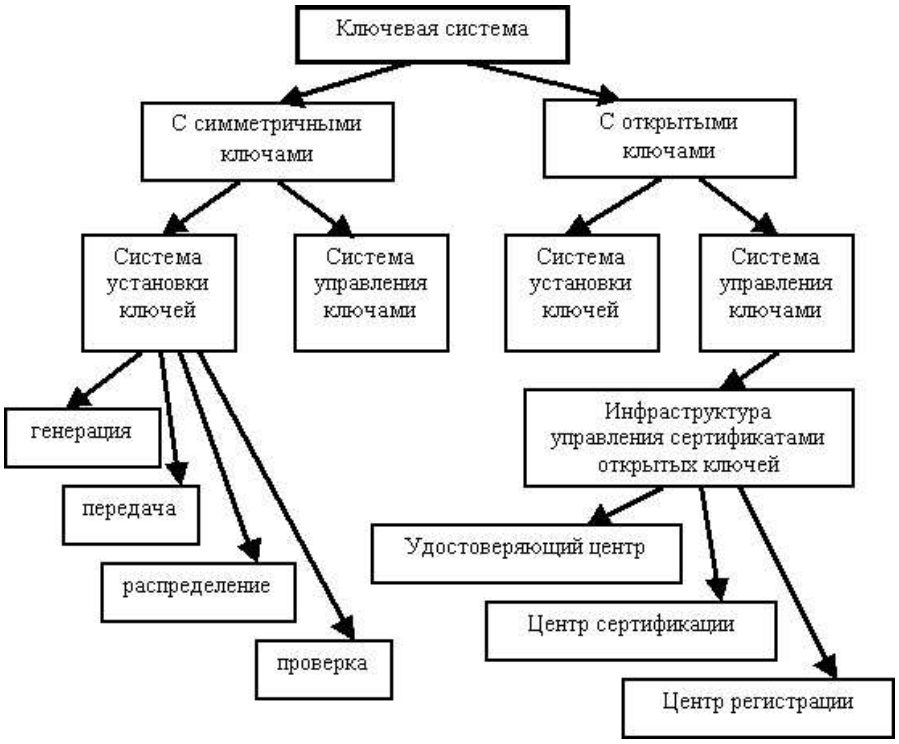


Рис. 3:

ключей и обеспечение их сохранности для сетей с большим числом абонентов.

Система ключевая асимметричной криптосистемы — основана на использовании асимметричных ключей, состоящих из пары — открытого и секретного (закрытого) ключей. Основными проблемами таких систем являются построение *системы управления ключами*, как правило, представляющей собой инфраструктуру управления сертификатами открытых ключей, включающую центры регистрации и сертификации. Функции обоих центров могут объединяться одном удостоверяющем центре.

Стойкость криптографическая — свойство криптографической системы, характеризующее ее способность противостоять атакам противника, как правило, с целью получить ключ, открытое сообщение или навязать ложное сообщение.

3 Элементы криптосистем

Алгоритм имитозащищающего кодирования информации — алгоритм преобразования информации (как правило, основан на внесении и использовании избыточности) с целью контроля *целостности*. В отличие от алгоритма формирования цифровой подписи, использует симметричные криптографические системы. В качестве такого преобразования может выступать *код аутентификации*, автоматное и другие преобразования, либо *алгоритм шифрования*.

Алгоритм проверки цифровой подписи — алгоритм, в качестве исходных данных которого используются подписанное сообщение, ключ проверки и параметры *схемы цифровой подписи*, а результатом является заключение о правильности или ошибочности *цифровой подписи*.

Алгоритм расшифрования — алгоритм, реализующий функцию расшифрования.

Алгоритм формирования цифровой подписи — алгоритм, в качестве исходных данных которого используются сообщение, ключ подписи и параметры *схемы цифровой подписи*, а в результате формируется *цифровая подпись*.

Алгоритм шифрования — алгоритм, реализующий *функцию* шифрования[1].

[1]В некоторых случаях шифр реализуется совокупностью некоторых преобразований (например, реализуемых алгоритмами DES, AES, ГОСТ и т. п.) с помощью так называемого режима шифрования.

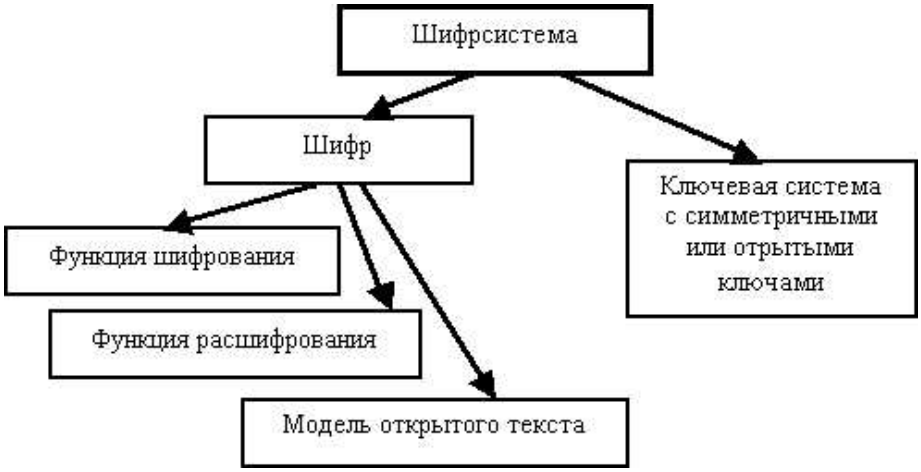


Рис. 4:

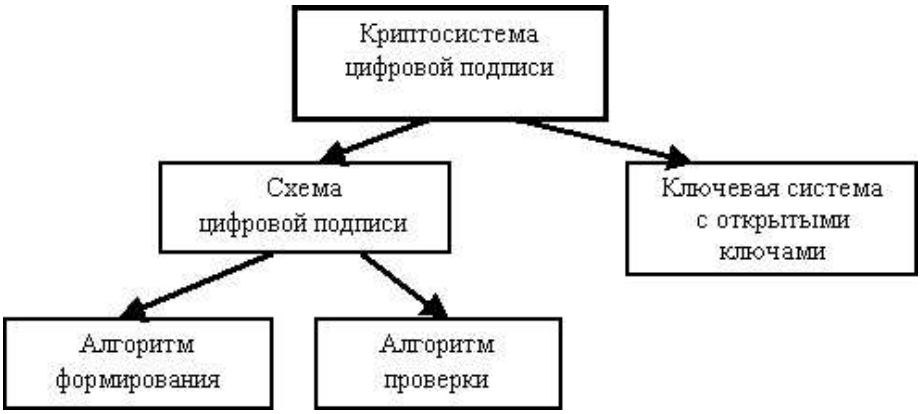


Рис. 5:

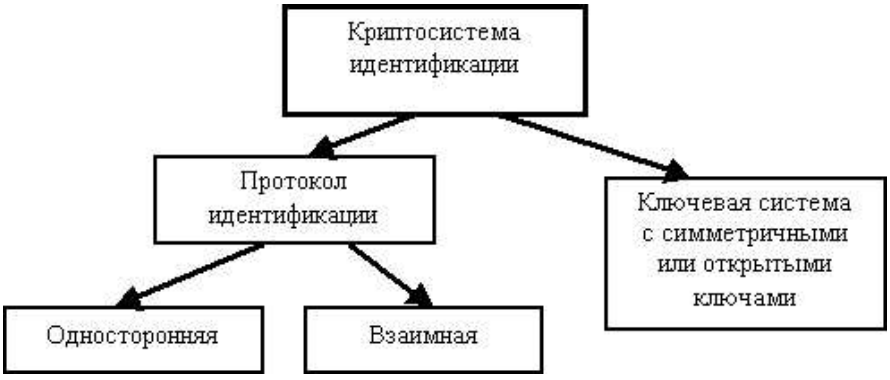


Рис. 6:

Жизненный цикл ключей — последовательность стадий, которые проходят ключи от момента генерации до уничтожения. Включает такие стадии, как: генерация ключей, регистрация пользователей и ключей, инициализация ключей, период действия, хранение ключа, замена ключа, архивирование, уничтожение ключей, восстановление ключей, отмена ключей.

Имитовставка — проверочная комбинация, добавляемая к сообщению для проверки целостности.

Имитостойкость — способность противостоять активным атакам со стороны противника, целью которых является навязывание ложного или подмена передаваемого сообщения или хранимых данных.

Код аутентификации — алгоритм имитозащищающего кодирования информации (как правило, вычисляет значение *имитовставки*). К кодам аутентификации предъявляются требования: большая сложность вычисления значения кода аутентификации для заданного сообщения без знания ключа; большая сложность подбора для заданного сообщения с известным значением кода аутентификации другого сообщения с известным значением кода аутентификации без знания ключа. Без знания секретного ключа вероятность успешного навязывания противником искаженной или ложной информации мала.

Открытое распределение ключей (согласование ключа, выработка общего значения ключа) — протокол, позволяющий двум абонентам выработать общий секретный ключ путем обмена сообщениями по открытому каналу связи без передачи какой-либо общей секретной информации, распределяемой заранее. Важным преимуществом открытого распределения является то, что ни один из абонентов заранее не может определить значение ключа, так как ключ зависит от сообщений, передаваемых в процессе обмена.

Помехоустойчивость — способность сохранять устойчивую работу при наличии помех в канале связи.

Протокол — распределенный алгоритм, в котором участвуют две или более стороны, обменивающиеся между собой сообщениями.

Протокол идентификации — протокол *аутентификации сторон*, участвующих во взаимодействии и не доверяющих друг другу. Различают протоколы односторонней и взаимной идентификации. Протоколы идентификации, как правило, основаны на известной обеим сторонам информации (пароли, личные идентификационные номера (PIN), ключи). В дополнение к протоколу идентификации могут использоваться некоторые физические приборы, с помощью которых и проводится идентификация (магнитная или интеллектуальная пластиковая карта, или прибор, генерирующий меняющиеся со временем пароли), а также физические параметры, составляющие неотъемлемую принадлежность доказывающего (подписи, отпечатки пальцев, характеристики голоса, геометрия руки и т. д.).

Протокол криптографический — протокол, предназначенный для выполнения функций криптографической системы, в процессе выполнения которого стороны используют *криптографические алгоритмы*.

Протокол распределения ключей — протокол, в результате выполнения которого взаимодействующие стороны (участники, группы участников) получают необходимые для функционирования криптографической системы ключи. Различают следующие типы протоколов распределения ключей: протоколы передачи (уже сгенерированных) ключей; протоколы (совместной) выработки общего ключа (*открытое распределение ключей*); *схемы предварительного распределения ключей*. В зависимости от порядка взаимодействия сторон выделяют *двусторонние протоколы*, в которых стороны осуществляют передачу ключей при непосредственном взаимодействии, или, иначе, протоколы типа «точка-точка», и *протоколы с централизованным распределением ключей*, предусматривающие наличие третьей стороны, играющей роль доверенного центра.

Схема цифровой подписи состоит из двух алгоритмов, один — для формирования, а второй — для проверки подписи. Надежность схемы цифровой подписи определяется сложностью следующих трех задач для лица, не являющегося владельцем секретного ключа: *подделки подписи*, то есть вычисления значения подписи под заданным документом; *создания подписанного сообщения*, то есть нахождения хотя бы одного сообщения с правильным значением подписи; *подмены сообщения*, то есть подбора двух различных сообщений с одинаковыми значениями подписи.

Схема предварительного распределения ключей — состоит из двух алгоритмов: распределения исходной ключевой информации и формирования ключа. С помощью первого алгоритма осуществляется генерация исходной ключевой информации. Эта информация включает открытую часть, которая будет передана всем сторонам или помещена на общедоступном сервере, а также секретные части каждой стороны. Второй алгоритм предназначен для вычисления действующего значения ключа для взаимодействия между абонентами по имеющейся у них секретной и общей открытой части исходной

ключевой информации. Применяется для уменьшения объема хранимой и распределяемой секретной ключевой информации. Схема предварительного распределения ключей должна быть устойчивой, то есть учитывать возможность раскрытия части ключей при компрометации, обмане или сговоре абонентов, и гибкой — допускать возможность быстрого восстановления путем исключения скомпрометированных и подключения новых абонентов.

Функция криптографическая — функция, необходимая для реализации *криптографической системы*, например, генерация ключей и псевдослучайных последовательностей, обратимое преобразование, однонаправленная функция, вычисление и проверка значений имитовставки и цифровой подписи, вычисление значения хэш-функции и т. п., обладают определенными криптографическими свойствами, влияющими на криптографическую стойкость: зависимость от ключа, сложность обращения и др.

Функция расшифрования — осуществляет преобразование множества открытых сообщений в множество шифрованных сообщений, зависящее от ключа, является обратным к преобразованию, осуществляемому *функцией шифрования*.

Функция шифрования — осуществляет преобразование множества открытых сообщений в множество шифрованных сообщений, зависящее от ключа.

Цифровая подпись (сообщения или электронного документа) — представляет собой конечную цифровую последовательность, зависящую от самого сообщения или документа и от секретного ключа, известного только подписывающему субъекту, предназначенная для установления авторства. Предполагается, что цифровая подпись должна быть легко проверяемой без получения доступа к секретному ключу. При возникновении спорной ситуации, связанной с отказом подписывающего от факта подписи некоторого сообщения либо с попыткой подделки подписи, третья сторона должна иметь возможность разрешить спор. Цифровая подпись позволяет решить следующие три задачи: осуществить *аутентификацию источника* данных, установить целостность сообщения или электронного документа, обеспечить невозможность отказа от факта подписи конкретного сообщения.

Шифр — семейство обратимых преобразований множества открытых сообщений в множество шифрованных сообщений и обратно, каждое из которых определяется некоторым параметром, называемым ключом. Математическая модель шифра включает две функции: *шифрования* и *расшифрования*, и модель множества открытых сообщений. В зависимости от способа представления открытых сообщений различают блочные, поточные и другие шифры. Основными требованиями, определяющими качество шифра, являются: *криптографическая стойкость*, *имитостойкость*, *помехоустойчивость* и др.

Литература

[1] Шеннон К. Теория связи в секретных системах. В кн.: Работы по теории информации и кибернетике. М.: ИЛ, 1963.

[2] Мэсси Жд. Л. Введение в современную криптологию. ТИИЭР, 1988, Т. 76, № 5, С. 24–42.

[3] Погорелов Б. А., Черемушкин А. В., Чечета С. И. К вопросу о терминологии, используемой в криптографии. Вестник Томского университета. Приложение. Материалы научных конференций, симпозиумов, школ, проводимых в ТГУ. 2003, № 6, 53–57.

[4] Menezes A. J., van Oorschot P. C., Vanstone S. A Handbook of applied cryptography. CRC Press, Boca Raton, New York, London, Tokyo, 1997.

[5] Stinson D. R. Cryptography: Theory and practice. CRC Press, N.Y., 1995.

[6] Алферов А. П, Зубов А. Ю., Кузьмин А. С., Черемушкин А. В. Основы криптографии. Учебное пособие. 2-е изд., доп. М.: Гелиос АРВ, 2002.

[7] Шнайер Б. Прикладная криптография. М.: Триумф, 2002.

Христиан Гольдбах и Франц Эпинус (из истории шифровальных служб России XVIII века)

В. К. Новик

Аннотация

Развитие любой сферы человеческой деятельности достигает только тогда своих высот, когда к этой сфере спонтанно или вынужденно подключаются лучшие ученые умы. Всегда и везде «кадры решают все». Период становления высокопрофессиональных шифровальных служб России (XVIII век) — прекрасная иллюстрация этой вечной истины. Академики Санкт-Петербургской АН России Христиан Гольдбах (1690–1764 гг.) и Франц Эпинус (1724–1802 гг.) возглавляли эту службу соответственно в 1742–1764 и 1765–1797 годах. В статье кратко излагаются обстоятельства появления их на этих постах и некоторые следствия их работы для истории России.

1 Введение

В России необходимость создания выделенных шифровальных служб была осознана на государственном уровне во времена Петра I, когда за рубежом оказалось множество дипломатических представительств и систематическая связь с ними должна была быть доверена почте, а не эпизодически посылаемым курьерам. Именно в Коллегии Иностранных Дел, а не в какой-то другой государственной Коллегии, Комиссии или Совете была порождена эта особая служба. Шифры этого времени были примитивны, часто иероглифические [1] и без труда читались в иностранных службах перлюстрации [2]. Россия не имела и еще не успела взрастить профессионалов, способных создать шифрсистемы, способные устоять перед лучшими криптографами Европы. И по той же причине русские службы перлюстрации должны были довольствоваться лишь перекупленными ключами. Бесконечная чехарда канцлеров в послепетровское время не могла, естественно, сколь либо укрепить эту сторону деятельности Коллегии Иностранных Дел.

Первым таким профессионалом сумел стать математик Христиан Гольдбах (Christian Goldbach, 1690–1764 гг.), вовлеченный в эту службу в 1742 г. и возглавивший ее в 1744 г.

Христиан Гольдбах занимал этот пост в различных чинах с 1744 по 1764 год. Сменил его Франц Эпинус (Franz Ulrich Theodosius Aepinus, 1724–1802 гг.), прослуживший на «особливой должности» с 1765 по 1797 год.

У них было удивительно много общего. Оба они были немцы, оба протестанты, оба родились на побережье Балтийского моря (Гольдбах — в г. Кенигсберге, Эпинус — в г. Ростоке). И тот, и другой были математиками (Эпинус еще и физиком), оба — полиглоты, оба — члены Санкт-Петербургской АН, оба оставили свой след и имя в истории науки. Оба были исключительно организованы и педантичны. Набор таких качеств отвечал минимальным требованиям к криптографу XVII века. Оба непосредственно сотрудничали с Великим Леонардом Эйлером (Leonhard Euler, 1707–1783 гг.). Оба были и равно наказаны судьбой — оба неженаты, оба через 15–17 лет после начала работы в КИД отмечены заболеваниями мозга, и оба скончались в состоянии психического расстройства. Их портреты до сих пор не найдены (да и сохранились ли они?), а личные архивы рассеяны по множеству хранилищ.

2 Время Христиана Гольдбаха

Биография Х. Гольдбаха и его деятельность на научном поприще в Санкт-Петербургской АН изложена, по доступным авторам документам, в книге [3]. Добавим к этим сведениям, что он был ранее также назван и первым русским журналистом [4].

Предопределенная, казалось бы, судьба профессора академии Х. Гольдбаха, претерпела крутой поворот весной 1742 г. Несколько ранее президент АН Карл фон Бреверн (Karl von Brewern, 1704–1744 гг.), под непосредственным началом которого Гольдбах работал с апреля 1740 г., был назначен главой КИД указом вступившей на престол 25 ноября 1741 г. (здесь и далее старый стиль) Елизаветы I. Карл фон Бреверн оказался руководителем с должным кругозором, способным осознать требования времени к персоналу шифровальной службы и, самое главное, требования к интеллекту, эрудиции и нравственности руководителя этой службы. Найти необходимого человека возможно лишь в том случае, если отчетливо понимаешь, каким он должен быть. Естественно, что необходимую персону, способную воспринять и самостоятельно развить столь специфический вид интеллектуальной деятельности, он мог извлечь только из математиков АН (в других местах таковых просто не было). И Бреверну посчастливилось увлечь такой работой Христиана Гольдбаха, единственного математика, оставшегося в Академии после отъезда летом 1741 года Л. Эйлера в Берлин.

Именным указом Императрицы от 18 марта 1742 г. коллежский советник [3, с. 83] Гольдбах был пожалован в статские советники и зачислен в КИД с окладом «для ободрения» в 1500 рублей в год. В АН с 1733 г. его оклад составлял 1000 руб. [3, с. 80]. Через месяц он писал своему старому знакомому: «... Я нашел свое место и свое назначение.» [3, с. 92], а в письме к Л. Эйлеру от 7 июня того же года он так высказывался о своем решении: «Зря не рискуй, но и не трусь» [3, с. 95]. Мы не знаем, было ли это рукой Провидения или нет, но именно в этом же письме Гольдбах сформулировал свою знаменитую гипотезу о том, что «...каждое число большее чем 2, есть сумма трех простых чисел» [3, с. 170]. Доказана ли она сейчас в общем виде?

Около года затратил Х. Гольдбах на приобретение практических навыков в новом деле составления шифров и дешифровки перехваченных депеш. Второе, естественно, более важно. В июле 1743 г. он представляет первую, аналитически полученную, дешифровку депеши «Австрийско-Цесарскаго министра в России» Нейгауза. К декабрю 1743 г. он прочитал уже 61 письмо прусского и французского послов [5]. Дееспособность Гольдбаха в новой службе уже не вызывала сомнений и 22 февраля 1744 г. с ним был подписан контракт о работе в КИД [6]:

«Да будет известно каждому и всякому, кому знать дозволено, что по Ея Российского Императорского Величества всемилостивейшему Указу, государственная Коллегия Иностранных Дел со статским советником Гольдбахом, в Ея Величества службе находящимся, ниже подписавшимся, следующий контракт заключен, а имянно:

То, что он, статский советник Гольдбах, по чаемому отпуску от Его Величества Короля Прусского, подданным которого он является, принимается здесь на службу, с тем, чтобы все то, что поручаемо ему его начальниками упомянутой Коллегии, станет, согласно его обязанности по присяге, с надлежащим усердием и крайним служебным рвением, как нам верноподданным Ея Императорского Величества надлежит, соблюдать и исполнять.

До тех пор будет по контракту его ежегодное жалованье тысяча пятьсот рублей, которое ему каждую указанную треть должно будет выплачиваться, и при том равными долями будет ему отдаваться.

1) чтобы он, статский советник был бы освобожден в вышеуказанной службе от ежедневного пребывания в Канцелярии упомянутой Коллегии, и сам там должен будет появляться, если только, он туда приглашен будет. Вместе с тем он требует, чтоб ему порученные дела в своей квартире исполнять для состояния лучшего удобства было можно,

2) Должен он, статский советник, все время лишь в этой своей службе в Коллегии Иностранных Дел проводить, и от этого ни в комиссии, ни на другие обязанности, ни мало не отвлекаться.

3) К отправке в чужие земли, вне Российской Империи опять же по его воле исполнено будет. В прочем же, однако, его обязанностью является, то, что поручаться ему его начальниками будет, все без исключения, безусловно исполнять со всевозможным старанием, также он, как кроме того, к тому, что от него до производимых дел касается малейшей безответственности опасаться должен, а скорее сим Ея Императорского Величества всевысочайшее покровительство уверить станет.

В подтверждение всего этого как от государственной Коллегии Иностранных Дел Президентом, так и от самого статского советника два равнозначных экземпляра этого контракта собственноручно подписаны, и из них один в той же Коллегии приобщен, а другой, однако, отдан статскому советнику.

В Москве, февраля 22 дня 1744 года.

По Ея Императорского Величества всемилостивейшему Указу Алексей граф Бестужев-Рюмин».

Отметим, применительно к этому контракту,

1) что его копия была снята будущим непременным секретарем Конференции АН Г. Ф. Миллером,

2) что Гольдбах выговорил право проводить секретнейшую работу на дому и

3) что общая сумма, выделяемая на деятельность КИД, составляла всего 28 000 рублей в год.

Успехи Гольдбаха в дешифровке депеш французского посла маркиза де Шетарди стали достоянием всех хрестоматий по истории криптографии. Зная, что его письма вскрываются, де Шетарди был уверен, что прочесть его шифр невозможно и поэтому легкомысленно писал об императрице, что она всецело отдается своим удовольствиям, легкомысленна, глупа и беспутна. Переводы этих 17 писем можно прочесть и сегодня [7]. 5 июня, ставший через месяц канцлером, вице-канцлер А. Бестужев-Рюмин разыграл перед Елизаветой сцену дешифровки депеши, «вынужденно» произнося «поносные слова». 6 июня де Шетарди был выдворен из страны. 26 июля Х. Гольдбах стал действительным статским советником [3, с. 96], что по табели о рангах соответствовало военному чину генерал-майора. С повышением в чине его не замедлил поздравить из Берлина стародавний товарищ и корреспондент Л. Эйлер [8], для которого не были секретом занятия Гольдбаха.

Дешифровки Гольдбаха значили многое — Россия на равных вступала в единоборство с лучшими умами в Черных Кабинетах Европы. На Гольдбаха посыпались всяческие изъявления довольства власть предержащих [3, с. 96], но мы отметим лишь одно — Двор и Императрица воочию ощутили, что математика и криптография для государства и их лично, это не нечто престижно-декоративное, а щит и меч, охраняющие их непосредственные интересы.

Конечно, «нам не дано предугадать, как слово наше отзовется...». Дешифровки писем де Шетарди вполне могли изменить историю России. Девятого февраля 1744 г. в Москву, где проистекали эти события, вместе со своей матерью княгиней Иоганной-Елизаветой Ангальт-Цербстской прибыла невеста наследника престола Софья-Августа-Фредерика. Екатерина Великая писала впоследствии, что в письмах де Шетарди «нашли... разсказы о неосторожных разговорах, которые он имел с моею матерью, что так сильно разсердило императрицу, что мой брак чуть от этого не разстроился... однажды утром граф Лесток вошел и сказал матери: «Готовьтесь уезжать» [9]. Россия, разумеется, должна благодарить Елизавету за благоразумную сдержанность, но пятнадцатилетняя девочка, впервые услышавшая при столь драматических обстоятельствах слово «шифр», очень хорошо усвоила этот урок.

Шифры, составленные Гольдбахом, сохранились до настоящего времени [10], и вполне доступны для исследований. И, пожалуй, наибольший научный интерес представили бы эти исследования, если бы в результате их удалось найти отражение в шифрах идей комбинаторики, интенсивно обсуждавшихся в 1740–50 годах в переписке Гольдбаха и Эйлера [11]. Отнюдь не случайность, что именно с 1742 г. Гольдбах увлекся теорией чисел, причем эта переписка на длительное время стала для него единственным каналом математических дискуссий [3, с. 101]. Любопытно и другое временное совпадение — в 1744 г. Л. Эйлер посылает одному из друзей наивную криптограмму, которую он считает неразрешимой [5, с. 118]. Налицо период взаимного пробуждения интересов к различным сторонам специфической, интеллектуальной сферы — криптографии.

Неотъемлемые черты такой сферы, секретность, постоянная настороженность и «умственная» напряженность обусловили и необычность его поведения в быту. Современник писал [12]: «... Гольдбах отличался большими странностями. Подобно Месмеру, он верил влиянию одного существ на другое и не выносил, чтобы к нему подходили ближе известнаго разстояния, чтобы ели и пили после него, или дотрагивались бы до чего-нибудь из того, к чему он прикасался какою-либо частью своего тела. Остатки того, что он пил и ел, он сам выбрасывал в окно. В конце года он пересматривал свой гардероб, и все ненужное сжигалось в нарочно разведенном огне. Можно представить себе, как он боялся заразиться, во время своей работы над депешами. Занимался он, с разрешения Императрицы, исключительно у себя в кабинете...». Другой иностранец, лично знакомый с Гольдбахом отмечал: «... Он бы человек большой учености, огромной эрудиции, большого ума и потрясающей памяти, благодаря которой он мог читать наизусть целые страницы из старых классических авторов...» и, выражая глубочайший пиетет, перечислял, бросающиеся в глаза, его безобидные чудачества [3, с. 105, 106].

В 1755 г. в Берлинской АН по приглашению Л. Эйлера должность астронома занял Ф. У. Т. Эпинус [13]. В условиях начавшейся семилетней войны, Эпинус в марте 1757 г. покинул Академию и при содействии Л. Эйлера переехал в Россию на должность профессора физики Санкт-Петербургской АН. С собой Эпинус увез рекомендательное письмо Эйлера, адресованное почему-то не кому-либо из действующих членов Санкт-Петербургской АН, а к Гольдбаху. В письме Эпинус характеризовался

некоей загадочной фразой: «...Г-н профессор Эпинус, который имеет честь вручить это письмо, является не только моим очень хорошим другом, но имеет кроме своей основательной учености такие заслуги, которые дают мне основание считать, что его знакомство с Вашим Высокоблагородием будет не неприятным, и поэтому я осмеливаюсь рекомендовать его Вашему Высокоблагородию...» [14]. Так пересеклись жизненные пути героев этой статьи.

Месяцем ранее Л. Эйлер писал Г. Ф. Миллеру, секретарю академического собрания Санкт-Петербургской АН и одному из немногочисленных друзей Гольдбаха, что выписал, по его просьбе, из Базеля работу Якоба Германа (Jacob Hermann, 1678–1733) по составлению шифров и дешифровке и, сообщая об ее пересылке, высказывает о ней свое мнение: «На этих днях я получил Приложенное из Базеля, и, хотя я не считаю содержание его сколь либо важным, я постараюсь переслать его Вашему Высокоблагородию. Идея писать скрытно таким способом, может быть, пожалуй, остроумной, и, вероятно, не может быть разгадана. Но когда тайнопись ⟨Geheimnüß⟩ как-нибудь случайно становится известной, сразу же исчезает вся польза для дальнейшего. Вполне возможно, однако, что я здесь и ошибаюсь, поскольку о тайнописи мне ничего не известно, и знаток, по-видимому, то же самое оценит крайне высоко, но я, по крайней мере, никогда не смог заметить, чтобы г-н профессор Герман, когда-либо прославил себя в этих делах большими и важными открытиями» [15].

Но уже в конце мая 1758 г. Эйлер получает еще одно письмо от наследника Я. Германа из Базеля с небескорыстным предложением подробно объяснить метод шифровки, изобретенный Я. Германом. К письму прилагались два листка таблиц с шифром [16]. Переписка с Россией затруднена, и Эйлер решает испытать стойкость шифра. Он предлагает Беглену (Nicolas de Bequelin, 1714–1789 гг.), директору Физического отделения Берлинской АН, вскрыть этот шифр. Тот, вопреки собственным сомнениям и к своему изумлению, всего лишь за три дня прочитывает его. Ценность шифра, естественно, уже не рассматривается, но появляется повод для сообщения в собрании Академии о процедуре дешифровки. 31 августа 1758 г. в очередном собрании Л. Эйлер зачитал мемуар Беглена «Об открытии одного шифра, предложенного покойным г-м профессором Германом, как абсолютно не раскрываемого» [17] (рис. 1, 2).

Складывающиеся обстоятельства вольно или невольно, но втягивали Л. Эйлера в шифровальные дела. И, как видно из дальнейшего, втянули таки — кратковременно, но всерьез.

14-го августа 1758 г. произошла битва под Цорндорфом, в которой русская армия, возглавляемая генералом В. В. Фермором (1702–1771 гг.) победила в сражении, понеся большие потери. Ровно через два месяца Л. Эйлер писал президенту Берлинской АН Мопертюи (Pierre Louis Moreau de Maupertuis, 1698–1759 гг.) в Париж: «Поскольку был перехвачен курьер, которого Фермор направил 25 сентября ко двору, мне было поручено изучить[1] все русские письма, число которых достигало нескольких сотен, и перевести те из них, которые могли бы что-либо осветить....» [18].

В подстрочном примечании составитель [18] отмечает: [1] Эйлер, перед тем как приехать в Берлин, служил профессором физики и математики в СПБ и, как знающий русский язык, он был, естественно, назначен для расшифровки ⟨в ориг. — déchiffrer⟩ русских депеш, о которых идет речь.

Пролить свет на интереснейший вопрос — Была ли это дешифровка или просто перевод писем с русского языка? — могло бы непосредственное обращение к этим документам, хранящимся, как утверждается в [19], в архиве ФРГ — DZA Merseburg, Rep. 63, n. 85 — Deutsches Zentralarchiv, Abt. II Merseburg (ранее ZStA: Zentrales Staatsarchiv der DDR, Dienststelle Merseburg).

Образец шифра (шифрант и дешифрант), данный генералу Фермору сохранился и доступен [20]. В нем каждой букве (слову, слогу) соответствует число от 0001 до 1050, т. е. шифруется 1051 литерная единица.

История с шифрами Я. Германа завершилась, повидимому, лишь в 1764 г., когда Эйлеру прислали еще два новейших типа таких шифров [21]. Укажем также, что образцы шифров, в том числе и Я. Германа, присланных в Санкт-Петербургскую АН, обсуждались и на ее заседаниях [22].

В сентябре того же 1758 г. в Санкт-Петербургской АН состоялось Публичное собрание, где, среди других, Франц Эпинус держал «Речь о сходстве електрической силы с магнитною» на латинском языке. Один из подносных ко Двору экземпляров «Речи», переплетенный в «серебряной» глазет, должен был быть лично вручен супруге наследника престола великой княгине Екатерине Алексеевне. И перед великой княгиней предстал остроумный молодой человек, известный ученый, к тому же медик, с которым можно было обсудить и собственные интимные проблемы, эрудит в естественных науках, философии, государственном праве, истории, говорящий на четырех языках. Небосклон Петербурга не блистал избытком интеллектуальных звезд, и неординарные натуры нашли друг друга. Молодому профессору Академии всемилостивейше дозволено стать учителем супруги наследника престола, ее наставником

Рис. 1: Леонард Эйлер в середине 1750-х годов.

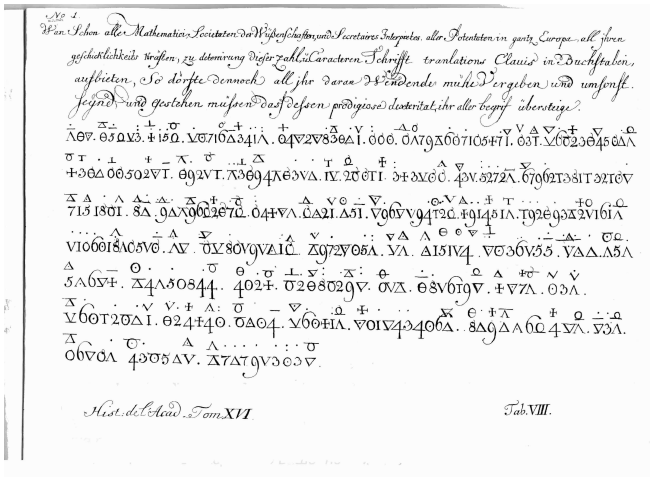


Рис. 2: Шифр Якоба Германа, представленный Л. Эйлером на обсуждение Королевской Прусской АН. В верхней части размещен озорной текст-призыв: «Уже ко всем математикам ученых обществ, ко всем секретарям-разборщикам всех властелинов целой Европы, ко всем силам их умения взываю вскрыть эти цифры и знаки и достоверно перевести в литерное письмо. Но так как, тем не менее, все ваше старание представить оригинал будет тщетным, то должно признать, что это необычайное искусство превосходит все ваше понимание.».

по физике, астрономии и даже в «математических знаниях». Современник отмечал [23]: «...будучи великой княгиней, она находила удовольствие в частых беседах с г-м Эпинусом о физике и астрономии и поручила ему написать для себя некоторое краткое изложение этих наук...». Великая княгиня не мыслила свою жизнь без престола, но столь же ясно она представляла требования к личному кругозору и интеллектуальным качествам самодержца. Конечно, образование призвано было дать не столько узкопрофессиональные знания, сколько представление о роли отдельных сфер человеческой деятельности в жизни государства. И, по этой причине, ее образование, точнее самообразование, не ограничивалось естественнонаучными дисциплинами, а включало историю, дипломатию, внутреннюю и внешнюю политику.

Осень 1758–весна 1759 года были черной полосой в биографии будущей Екатерины Великой. Несколько ранее были арестованы ближайшие к ней лица: сам А. П. Бестужев-Рюмин, В. Е. Ададуров, И. П. Елагин (1725–1794 гг.), ювелир Бернарди. Регулярное общение с опальной, одинокой и всеми заброшенной княгиней было не безопасно и требовало прозорливости в предопределении ее судьбы. Возможно, этим и оправданы строки Эпинуса в предисловии к написанному весной 1759 г. эссе [24] (издано в 1771 г.) под названием «Разсуждение о строении мира», что сочинение «...оное ему впервые отверзло путь к милостям...государыни...в чем он за несколько лет предупредил ту рода человеческого часть, с которою вместе жил...». Блистательные успехи ученицы дают его предвидению самый лестный аттестат.

Причины отдаления великой княгини были зримы и наглядны — неудавшееся Бестужеву-Рюмину устранение великого князя от права наследования с назначением таковым малолетнего Павла Петровича при регентстве Екатерины. Лишь величайшая дипломатическая изворотливость сохранила ее при дворе. Уже несколько лет она была платным агентом английского посла [25], секретарь и шифровальщик которого Станислав Понятовский сделался ее любовником. Супруга наследника престола к этому времени в совершенстве владела навыками анализа и отбора информации, конспиративной связи и переписки, транспорта денег и привлечения единомышленников. По вступлении на престол выяснилось, что в переписке с Понятовским она использовала, как минимум, несколько шифров [26].

В ее окружении после ареста И. П. Елагина неизвестны лица, способные вести столь ответственную работу. Остается поверить, что она вела ее сама.

Ее характеристика в будущем — «Императрица Екатерина II блестяще владела навыками разведывательной и контрразведывательной работы» — могла бы быть сочтена досужей лестью Великой Государыне, если бы не была произнесена человеком, отдавшим разведке 35 лет жизни и 12 лет руководившим нелегальной разведкой КГБ СССР [27]. Что может быть объективнее, чем оценка потомков?

Начало 1759-го года стало роковым и для престарелого Христиана Гольдбаха. 1-го февраля он с тревогой сообщал вице-канцлеру М. И. Воронцову [28]: «Мсье, сегодня утром, когда я приготовился, чтобы подняться в карету, дабы отправиться к Вашему Сиятельству, со мной случилось что-то вроде головокружения, которое меня пугало еще два дня назад. Я было думал, что свалюсь на землю, но удержался, и я надеюсь вылечиться от этого несчастья, которое могло бы легко перейти в апоплексический припадок, приемлемыми средствами и строгой диетой...».

Это был первый сигнал о заболевании мозга, а упование на диету в комментариях не нуждается. 16 августа 1760 г. Гольдбах был пожалован в тайные советники с окладом 3000 руб. в год [29]. Его тягостное недомогание перешло в тяжелое заболевание, по-видимому, в болезнь Паркинсона. 20-го ноября 1764 г. он скончался в возрасте 74-х лет. И Франц Эпинус был обречен [13] его заменить.

3 Время Франца Эпинуса

Достаточно целостное представление о работе шифровальной службы КИД этого периода дано в [5, с. 124 и далее], [13]. Поэтому остановимся на некоторых частных моментах, важных для ее деятельности, нашедших отражение в сохранившихся документах.

И первое, о чем должно высказать мнение, — это о способе приобретения необходимых профессиональных навыков. Ни Гольдбах, ни Эпинус целенаправленно не получали какой-либо специальной подготовки в этой экзотической сфере. Так что очевидной является их способность к самообразованию, опирающаяся на приобретенную ранее математическую эрудицию и общую интеллектуальную культуру. Развитие математической эрудиции, да и логики мышления, в свою очередь, побуждалось, главным образом, лишь воспитанием, воспринятыми жизненными ценностями, примерами ярких личностей, приобщением к такому кругу задач и людей, готовностью к такого рода труду. Перечисленные качества уже сами по себе в XVIII веке были свойственны даже не единицам, а исключительностям.

Пример общей математической эрудиции. Через месяц после приезда в Петербург, едва решив дела устройства в новой службе в чужой пока что стране, Эпинус через три границы пишет Эйлеру: «...Я очень благодарен Вам за разъяснение моего сомнения о вычислении вероятностей. Во всяком случае, разные выражения, встречающиеся у Бернулли, ввели меня в заблуждение, и я путал “valorem expectationis” (*значение ожидания*) с “mensura absoluta probabilitatis” (*абсолютная мера вероятности*). Теперь я, во всяком случае, знаю, что правильное различие этих понятий снимает мое сомнение...» [30]. Эта цитата важна как свидетельство и фанатичного интереса к математике, и всестороннего критического осмысления классического труда Я. Бернулли «ARS CONJECTANDI (Искусство догадки)» [31]. Криптографам не нужно разъяснять прикладное значение этого труда, но ведь изучался то он сугубо с академических позиций.

Основой самообразования являются книга и общение. В каталоге библиотеки Эпинуса [13] указана книга, по сути учебник по криптографии, автора, скрывшегося под псевдонимом Густава Селена (Gustavus Selenus) (рис. 3). Для Эпинуса псевдоним не представлял секрета. Еще помогая брату в делах Университетской библиотеки в г. Ростоке, он держал в руках книгу с дарственной надписью автора, герцога соседнего Брауншвейга, со столицей в Люнебурге. Коронованные особы считали за честь создавать такого рода творения. Как показывает врезка на рис. 3, книга была редкой даже в XVIII веке. Только существенная необходимость в ее использовании и повлекла ее приобретение, ибо лишь по этому принципу Эпинусом и комплектовалась библиотека.

В самообразовании и самообучении обязательно должны присутствовать некие учебные упражнения, в создании шифров — некие черновики, эскизы. Сохранился документ [32] (рис. 4), который можно отнести именно к категории шифровальных этюдов. Этюд был прочитан и прокомментирован в [33]. Содержание документа представляет собой набор не связанных по смыслу французских фраз, в которых двузначным числом шифруется каждая буква. Двойные штрихи разделяют слова. В заключительной части документа каждой букве ставится в соответствие уже пять двузначных чисел

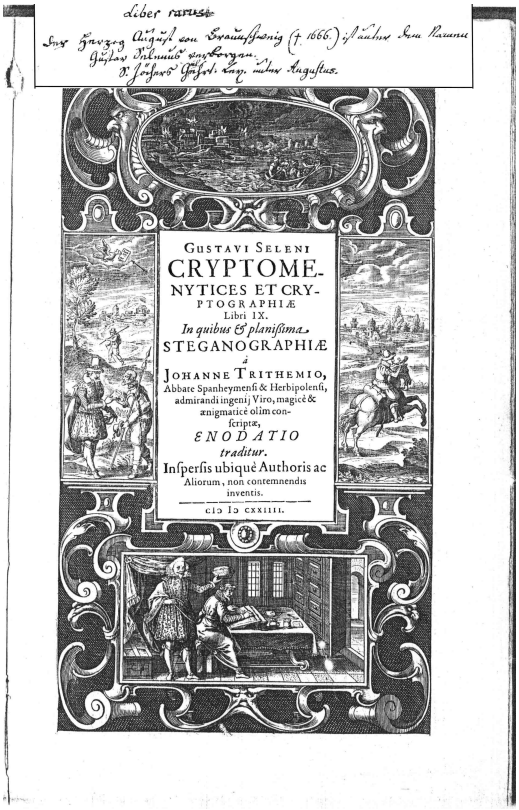


Рис. 3: Титульный лист книги Густава Селена. Заглавие: «Девять книг Густава Селена о сокрытии смысла и тайнописи, в которых изъясняется учение о секретной переписке, встарь изложенное Иоганном Тритемием, настоятелем в Спенхейме и Хербиполене, мужем редкостного магического дара разгадывания. Здесь же собраны создатели также и других не менее важных открытий. 1624.». Фраза на врезке сверху: «Книга редкая. Герцог Август из Брауншвейга (умер в 1666 г.) является автором, скрывающимся под именем Густава Селена. См. ученый лексикон Йохера на «Август».»

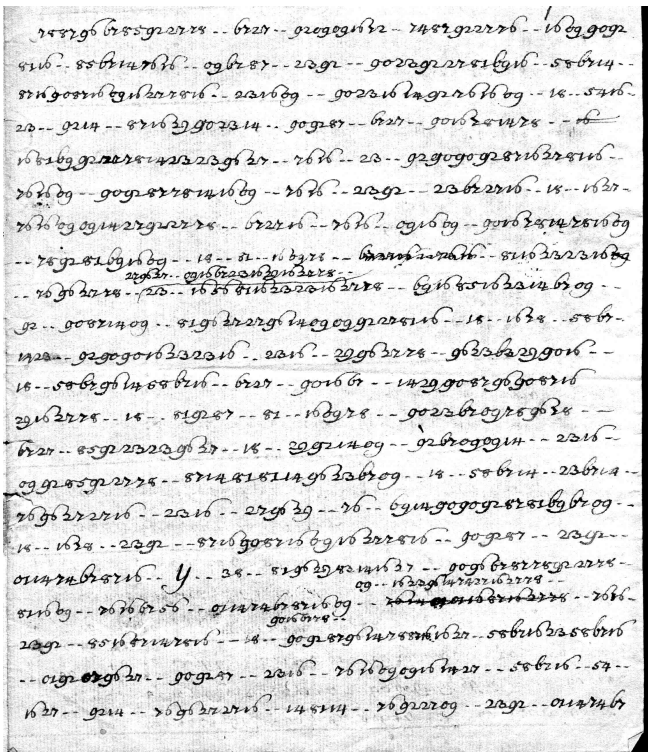


Рис. 4: Начальная страница шифровального этюда Франца Эпинуса.

со слитным, без пробелов, написанием. Шифр прост, и предназначен или для пробной обратной дешифровки, или, скорее всего, для дальнейшего совершенствования. Последняя фраза: «...пока сейчас с меня довольно, первая идея, что у меня появилась, была не лучшей, но...». Теперь этот автограф ценен как перечень фактов, позволяющих датировать документ и пролить свет на некоторые штрихи биографии Эпинуса.

Скучная рутинная практика ручной шифровки и дешифровки персоналом сотен страниц депеш, в условиях далеко не комфортных, автоматически отсеивала все новации, диктующие повышения требований к персоналу. Устоявшейся формой представления шифров были два типа таблиц. Первые — жестко ставящие в соответствие буквам, слогам, слогосочетаниям, словам и словосочетаниям четырехзначные (как правило) числа. Их называли шифрантами. Вторые — естественной последовательности четырехзначных чисел ставили в соответствие буквы, слоги и т. д. Их называли дешифрантами. Сопряженная пара «шифрант-дешифрант» и была тем, что подразумевалось под словом шифр или «цифирь». Для удобства чтения при плохой освещенности размеры букв и цифр должны были составлять 4–5 миллиметров, что, в конечном итоге, превращало «цифири» в толстые тетради или альбомы и вынуждало пользоваться ими годами и десятилетиями. Доставка нового шифра многочисленным представительствам за рубежом была весьма трудоемка. Длительность пользования — способствовала попаданию его копии к противнику.

Жизнь настоятельно требовала создания динамично перестраиваемой шифрсистемы, устойчивой к фактам предательства, к вскрытию по известному скрытому тексту и т. д.

Эпинус предложил такую систему [34], которая вполне достойна самостоятельного описания и анализа, хотя бы с методических позиций обсуждения принципов, положенных в ее основу, и детально аргументированных им. Здесь, однако, ограничимся ее краткой общей характеристикой.

Система была целиком ориентирована на аналитическую шифровку без участия таблиц. В нее закладывалось двух-, а при необходимости и трехэтапное шифрование. На первом этапе шифрование осуществлялось секретарем. Каждая буква скрывалась за одним из пяти двузначных чисел, после чего уже цифровой текст перешифровывался по оговоренной строке известной книги. На втором этапе этот цифровой текст еще раз перешифровывался начальником канцелярии или посланником, с использованием только ему известного восьмиразрядного числа. Иными словами, шифр представлял собой не что иное как жесткий алгоритм, в который вводились на разделенных этапах две или три, если перешифровывали последовательно и начальник канцелярии, и посланник, некоррелированных переменных — оговоренная книга и известные числа.

О преимуществах этого шифра Эпинус писал [34, л. 18]: «Этот шифр никогда не может быть расшифрован прямо, т. е. на основании комбинационной теории, как это обычно случается с литерными или другими плохо составленными шифрами. Его также невозможно вскрыть при перехвате, т. е. когда удается заполучить дешифровку одной или нескольких написанных этим шифром депеш... Если предположить, что секретарь, занимающийся шифровкой и дешифровкой, продаст этот шифр во всем его объеме, то из этого ничего не последует, ибо депеши, которые будут впоследствии писаться этим шифром, будут столь же непонятными и никоим образом не могут быть дешифрованы... Нужен только этот единственный шифр, чтобы можно было переписываться с любым количеством людей, не опасаясь, что кто-то поймет депеши, адресованные другому, ибо достаточно с каждым из них определить переменные элементы... Отсюда следует, что можно будет избежать частой, многотрудной и дорогостоящей посылки курьеров для передачи новых шифров или когда не хотят депеши большой важности подвергать опасностям и доверять почте... Этот шифр полностью отвечает требованию, выдвигаемому каждым, кто ставит перед собой задачу иметь не только стойкий, но и совершенно нераскрываемый шифр, а именно: «найти такой шифр, чтобы все встречающиеся в нем числа или знаки имели совершенно одинаковые вероятности, и этим шифром можно было бы писать что угодно». Но эта его особенность, и другие, не могут быть описаны здесь, а являются предметом чисто математического исследования, где и будет показано построение этого шифра на основе искусства догадки ⟨Art de conjecturer⟩ и комбинационной теории».

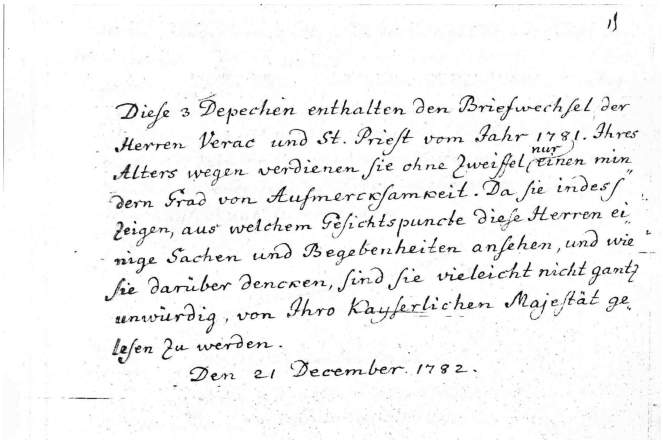
Как видим, штудирование книги Я. Бернулли не пропало втуне! Добавим, однако, и ложку дегтя. Шифр был перегружен вычислительными операциями, проводившимися вручную, и требовал качественного изменения квалификации ВСЕХ участников процесса. Именно это, по-видимому, и пресекло его распространение, так что табличные шифры были в ходу, во всяком случае, до середины XIX века.

В статье появилось слово «квалификация», так что затронем положение сотрудников КИД и шифровальщиков в «златой век Екатерины».

КИД была на острие «борьбы умов» и потому пестовала свои кадры, тщательно подбирая чиновников: «... Дела в Коллегии иностранных дел... суть... наиважнейшие», служители должны быть «умными и в делах... обученными, и вследствие их малолюдства принуждены будут работать день и ночь», поэтому «необходимо потребно видится оным учинить порядок , основательный и пропитание честное и довольное». При поступлении на работу в Коллегию все сдавали «дипломатический экзамен» — переводы с французского, немецкого и латинского на русский и с русского на немецкий и французский. Во всяком случае, сотрудники КИД были «совсем иные люди, чем приказные других мест». При Эпинусе и позже сотрудниками КИД были И. Ф. Богданович — автор «Душеньки», переводчик в Дрездене; писатель и архитектор Н. А. Львов; поэты Я. Б. Княжнин и В. В. Капнист. И, конечно, самый знаменитый и поныне из писателей «Века Екатерины» Денис Иванович Фонвизин, секретарь и шифровальщик «первоприсутствующего в Коллегии» графа Н. И. Панина.

Конечно, просто по необходимости, этот персонал обязан был стать интеллектуальной элитой страны. И на этом блистательном фоне особо, в силу своих редких качеств, выделялись криптографы. Они были «штучным товаром». Как и все остальные, они получали денежные награды, кольца и табакерки с бриллиантами, деревни и «людишек», но, в отличие от всех, им была дана редкостная привилегия прямого обращения к Императрице. И в своих обращениях они находили полную поддержку и понимание «великой жены», лично прикоснувшейся в свое время к их труду.

Сохранившиеся объемы дипломатической переписки, дешифрованной Эпинусом, воистину огромны. Это сотни и сотни листов, сплетенных в толстые фолианты. Далеко не всегда удавалось


15

Dieſe 3 Depechen enthalten den Briefwechſel der Herren Verac und St. Prieſt vom Jahr 1781. Ihres Alters wegen verdienen ſie ohne Zweifel nur einen min dern Grad von Aufmerckſamkeit. Da ſie indeſſ zeigen, aus welchem Geſichtspuncte dieſe Herren ei nige Sachen und Begebenheiten anſehen, und wie ſie darüber dencken, ſind ſie vieleicht nicht ganz unwürdig, von Ihro Kayſerlichen Majeſtät ge leſen zu werden.

Den 21 December 1782.

Рис. 5: Типичная сопроводительная записка Франца Эпинуса, прилагавшаяся к дешифрованным депешам, передаваемым Императрице: «Эти три депеши содержат переписку господ Верака и Сент-Притса за 1781 г. Из-за своей давности они, без сомнения, заслуживают минимального внимания. Однако, поскольку они указывают, с какой точки зрения эти господа рассматривают некоторые дела и события, и что они думают об этом, депеши, пожалуй, не являются совершенно недостойными того, чтобы не быть прочитанными Вашим Императорским Величеством. 21 декабря 1782 г.».

ему прочитать тексты целиком — имеются пропуски, доступными порой оказывались лишь отдельные абзацы или даже строки. Прочитанные депеши передавались Императрице с сопроводительными записочками (рис. 5) [35]. Прочитав депеши, Императрица отсылала их назад (рис. 6) [36]. Записочки интересны отсутствием обращения и подписи. Их содержание, стиль и тон соответствует скорее не отношениям сюзерена и подданного, а взаимно уважающих партнеров.

Разносторонняя многолетняя дипломатическая информированность Эпинуса должна была обусловить и привлечение его к выработке политическим решений. Наиболее известным из них является подготовка первой редакции знаменитой «Декларации о вооруженном нейтралитете», сыгравшей принципиальную роль в борьбе североамериканских колоний Англии за свою независимость [13], что и стало причиной постоянного его упоминания в работах по истории США.

Не умаляя частных заслуг криптографов России XVIII века, назовем главное: наша страна именно им обязана историческим событием непреходящей важности — созданием государственной системы народного просвещения.

На рис. 7 представлены две последние страницы [37] написанной собственноручно для себя Екатериной II памятной записки. Записка отражает итоги обсуждения с Эпинусом плана создания государственной системы народных школ. Событие это состоялось на первой неделе марта 1781 года. Заключительная строка записки сообщает: «...а вот тое все что с Эпинусом слóжили».

Это обсуждение предопределило рывок в создании и последующем развитии интеллектуального потенциала страны. Здесь нет преувеличения. Любая другая дефиниция значимости создания государственной системы просвещения будет слишком блеклой и обедненной. Речь идет именно о системе с едиными возрастами и сроками обучения, едиными учебниками и методиками, едиными программами обучения школьников и подготовки учителей.

Беседа знаменовала начало практических организационных работ, завершившихся созданием «Комиссии об учреждении народных училищ» (В. П. Завадовский, Ф. У. Т. Эпинус, П. И. Пастухов, первое заседание 13 сентября 1782 г.) с одновременным прессингом на общественное сознание (премьера «Недоросля» 24 сентября 1782г.), ярко подчеркивающим своевременность благодеяний Императрицы. При непосредственном участии Комиссии (в несколько расширенном составе) Е. Р. Дашкова была поставлена во главе Академии Наук. Комиссии же был поручен надзор за делами Московского Уни-

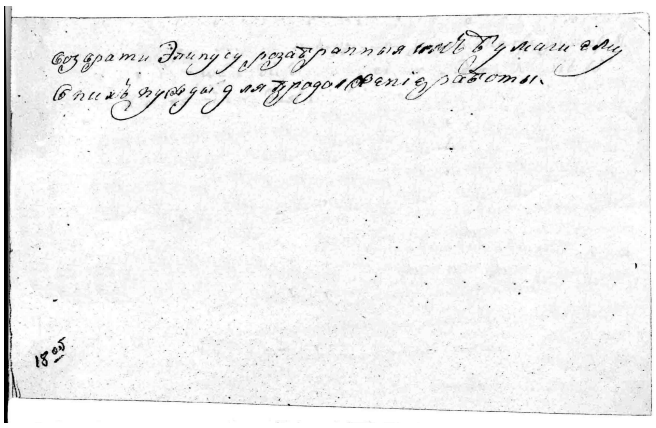


Рис. 6: Резолюция Екатерины II: «Возврати Эпинусу розабранныя имъ бумаги ему в нихъ нужды для продолжение работы.».

верситета, когда усилиями И. И. Шувалова началось возведение его зданий на Моховой. «Упорным рачением» Комиссии к 1791 году были созданы школы во всех 41-ой губернии и «земле донских казаков» — Россия встала в строй держав с государственной системой просвещения. Эпинус пробыл в составе Комиссии до своего ухода с государственной службы в 1797 году. Кто еще в нашей истории может написать в своей биографии строку — «...создал систему просвещения великого народа великой страны...»? Эта система, творчески развитая в СССР (Н. К. Крупская), до недавних пор была лучшей в мире.

С шифровальной службой КИД связан жизненный путь первого директора, первого в России Педагогического института, впоследствии преобразованного в Санкт-Петербургский Университет, Ивана Ивановича Коха (Johann Georg Koch, 1739–1805). И. И. Кох, по национальности венгр, родился в мещанской семье в местечке Гирнберг (Girnberg, Georgenberg), около города Касмарк (Käsemarkt, Käsmark) в волости Зипс (Zips) в восточном предгорье Высоких Татр. В 1762–1765 гг. служил копиистом и канцеляристом в Собрании Санкт-Петербургской АН. В 1766 г. был привлечен Эпинусом в КИД и стал его помощником в дешифровке перлюстрированной корреспонденции. С началом деятельности Комиссии был Эпинусом переведен и туда, где занимался организацией переводов и изданием учебников для школ и университетов. В начале 1787 г. И. И. Кох был назначен директором учительской семинарии, отделившейся к этому времени от главного народного училища. В 1803 году учительская семинария была переименована в учительскую гимназию, а в 1804 г. преобразована в Педагогический Институт, в должности директора которого Кох и завершил свою жизнь.

Какими же темпами впоследствии расцветало просвещение на просторах огромной Империи? Поскольку в конце XVIII века в школах обучалось всего лишь ~20000 детей, то экспертная оценка указывает на число грамотных ~2% всего населения страны. Что же изменилось в последующем столетии? «Если в годы перед реформами Александра II в России было только 6% грамотных, то к началу XX в. около 25% сельского и 45% городского населения умели читать и писать». И об этом сообщается как о величайшем достижении [38]. Потребовалась Октябрьская Революция, чтобы в середине 20-х годов началась повсеместная ликвидация безграмотности, а к 1934 году были созданы все условия для выполнения вводимого закона о ВСЕОБЩЕМ ОБЯЗАТЕЛЬНОМ начальном образовании.

Приведенные цифры позволяют ответить на весьма болезненный вопрос об отсутствии или, в лучшем случае, вторых ролях русских криптографов в XVIII веке. В стране просто-напросто не было достаточной исходной массы русских образованных людей, из которых можно было бы выделить криптографов со всей совокупностью их качеств, описанных ранее применительно к фигурам Гольдбаха и Эпинуса. Именно с совокупностью качеств, хотя русские «алгебраисты», потенциально пригодные для такой службы, были хорошо известны за рубежом [39].

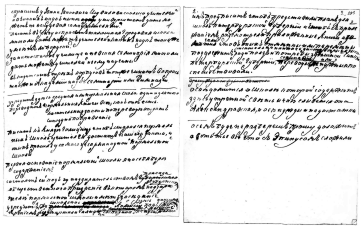


Рис. 7: Завершающий фрагмент записки Екатерины II (автограф) об учреждении школ. Ее текст, орфография сохранена: «...спросить у Ивана Ивановича Шувалова сколько учителей и в каких наук ныне университет дать может не оскудевая сама учителями?
писать в ригу и спрасить тамошная городская школа много ли дать может учителей и каких наук также умеють ли поруски?
заиконоспаское училище и невская семинария много ли дать может учителей и чему научаны в кадетских трех корпусов тотже чинит вопрос также и Академии наук семинарии или Гимназии.

учредит здесь средная или нормальная скола одна и для того переводит нормальныя книги все, как оне есть. Касательныя законы ⟨веры, религии⟩ по переводы отдать синоду на поправление
писать к Князю Голицину чтобъ старался нормальныхъ школь учителей достатъ нашего закона ⟨православных⟩, и оных просилъ у самаго цесаря на одной нормальной школы
первая основание нормальной школы я на себя берю содержаниемъ
состоятъ ей подъ надзирательством ~~губернскаго~~ приказа общественного призрение в которомъ губернскаму надзирателю нормальной школы иметь заседание
учредитъ же школьное Комисию здесь подъ смотрениемъ фелдмаршала Галицина т. с. ⟨тайного советника⟩ Мелесине господина Эпинуса Паласа глазами
имъ предписатъ чтобъ представили планъ для школъ питербурхской губернии и чтобы съ прилежаниемъ разсмотрели приложенные книги и зачали что въ оныхъ отменны или перемены поддлежитъ и касательна наипаче закона веры и языка ради ползы и потребы здешной санкт петербурхской губернии, сочинении или переводы же нужныя поспешествовали.

~~в рассуждении языков въ школе~~
осведомится о школе которой содержютъ здесь у тренной светь ⟨?⟩ и чаю слывется она Александровская, и о ею нужды и недостатки
о сем буде не разберешъ прошу доложитъ а вот и тое все что съ Эпинусомъ сло́жили».

Проблема «русские — немцы» особенно заостряется при взгляде на рис. 7, который дает недоброжелательному взгляду все основания заметить, что именно немка и пришлый немец принесли свет образования великому русскому народу. И это совершенно конкретные личности, а не мифические Кирилл и Мефодий.

Императрица, к тому времени, из своих 52-х лет 37 прожила в России, искренне, публично заявляя: «Я обязана России всем, даже именем». И совсем не случайно писала она в августе 1776 г. сыну Павлу: «Признаюсь чистосердечно, что самолюбию моему льстит безмерно честь не упадающего в мире русского имени». Эта немка была воистину Великой РУССКОЙ царицей. Так менталитет какого народа стал ей свойственен?

Эпинус из своих 57 лет уже 24 самых зрелых года жил в стране, где и стяжал мировую научную известность и мнение, что «он как никто хорошо знает Россию».

Так кто они были — русские или немцы? Для России этот вопрос не нов. И давным-давно В. Далем был дан на него весомый, краткий и афористичный ответ: «Ни прозвание, ни вероисповедание, ни самая кровь предков не делают человека принадлежностью той или иной народности. Дух, душа человека — вот где надо искать принадлежность его к тому или другому народу. Чем же можно определить принадлежность духа? Конечно, проявлением духа — мыслью. Кто на каком языке думает, тот к тому народу и принадлежит». Записка Екатериной II написана по-русски, значит, о России они думали по-русски!

Литература

[1] Подъяпольская Е. П. Шифрованная переписка в России в первой четверти XVIII века. В сб.: Проблемы источниковедения. М.: Изд. АН СССР, 1959, т. 8. с. 314–342.

[2] Kahn D. The codebreakers. The story of Secret Writing. NY: The Macmillan Co., 1968, p. 614–671, 1076–1086.

[3] Юшкевич А. П., Копелевич Ю. Х. Христиан Гольдбах. М.: Наука, 1983, 224 с.

[4] Булгаков А. Я. Ответ на библиографический вопрос. Московский телеграф, 1827, № 13, июль, с. 33.

[5] Соболева Т. А. Тайнопись в истории России. М.: Изд. «Международные отношения», 1994, с. 106, 107.

[6] РГАДА. Ф. 199. Оп. 1. Портфель. 247. Д. № 3 12. Л. 1–2. «Условие, данное Христианом Гольдбахом при определении его в Государственную Коллегию Иностранных Дел 22 февраля 1744 г.». Перевод с немецкого.

[7] АВПРИ. Ф. «Секретнейшие дела (перлюстрации)». Оп. 6/1. Д. № 20 «1744. Июня 6. Перечень (на французском языке с переводами) с писем французскаго в России министра маркиза Шетарди к разным французским министрам, прочтенный ему маркизу в Москве при объявлении ему удалиться в 24 часа из России за окорбительныя об Императрице и о Министрах Российских слова». Л. 1–8об.

[8] Леонард Эйлер. Переписка. Аннотированный указатель. Л.: Наука, 1967, с. 93. Письмо № 681. Берлин, Л. Эйлер — Х. Гольдбаху, 19 сентября 1744 г.

[9] Записки императрицы Екатерины Второй. СПб.: изд. А. С. Суворина, 1907, с. 478, 479.

[10] АВПРИ. Фонд «Секретнейшие дела (перлюстрации)», Оп. 3 173/1. Д. № 1. Л. 170, № 17.

[11] Малых А. А. Комбинаторный анализ в его развитии. Автореф. докт. дисс. М.: 1992, с. 14, 15.

[12] Анекдоты прошлаго столетия. Русский Архив, 1877, № 11, с. 288, 289.

[13] Новик В.К. Франц Эпинус (краткая биографическая хроника 1724–1802 гг.). Вопросы истории естествознания и техники, 1999, № 4, с. 4–35.

[14] Leonard Euler und Christian Goldbach, Briefwechsel 1729–1764. Berlin: Akad. Verl, 1965, S. 393.

[15] Серия "Quellen und Studien zur Geschichte Osteuropa". B. 3. Die Berliner und die Petersburger Akademie der Wissenschaften im Briefwechsel Leonhard Eulers. Teil 1. Der Briefwechsel L. Eulers mit G. F. Müller 1735–1767. Akad. Verlag. Berlin: 1959, S. 140. Письмо Л. Эйлера к Миллеру от 5 апреля 1757 г. Оригинал письма: СПбФ АРАН. Ф. 21. Оп. 3. Д. 321. Л. 42.

[16] Леонард Эйлер. Переписка (аннотированный указатель). Л.: Наука, 1967, № 585. Оригинал письма: СПбФ АРАН. Ф. 136. Оп. 2. Д. 5. Л. 226–228об.

[17] Winter E. Die Registres der Berliner Akademie der Wissenschaften 1746–1766. Berlin: Akad. Verl., 1957, S. 241.

[18] Le Sueur A. A. A. Maupertuis et ses correspondants: lettres inédits du grand Frédéric, du prince Henri de Prusse, de Labeaumelle, du président Henault, du comte de Tressan, d'Euler, de Kaestner, de Koenig, de Haller, de Condillac, de l'abbé d'Olivet, du maréchal d'Écosse etc. etc. etc. Montreuil-sur-mer : Impr. Nôtre-Dame des Prés, 1896. 448 p. pp. 158, 159.

[19] Leonhardi Euleri. Opera Omnia. Series Quarta A: Commercium Epistolicum. Vol. VI "Correspondaence between Euler and P.-L.-M. de Maupertuis". Basel: Birkhäuser Verlag, 1986. p. 285.

[20] АВПРИ. Ф. «Шифровальный отдел». Оп. 480/3. Д. 5174. «Шифр для переписки с графом Фермором 1759 г., 6/17 января». Л. 1–5об.

[21] Леонард Эйлер. Переписка (аннотированный указатель). Л.: Наука, 1967, № 586. Оригинал письма: СПбФ АРАН. Ф. 136. Оп. 2. Д. 5. Л. 229–229об.

[22] Протоколы заседаний конференции Императорской Академии Наук с 1725 по 1803 года. Том II. 1744–1770. СПб.: 1899, стр. 438.

[23] de la Marche C. Z. S. Anecdotes Russes ou Lettres d'un officier Allemand a un Gentilhomme Livonien, écrites de Petersbourg en 1762, Londres: 1764, p. 88, 89.

[24] РГАДА РФ. Ф. 181. Оп. 16. Д. № 1384.

[25] Переписка великой княгини Екатерины Алексеевны и английскаго посла сэра Ч. Г. Уилльямса, 1756 и 1757 г. С предисловием С. М. Горяинова. М.: 1909, с. 92.

[26] Записки императрицы Екатерины Второй, СПб.: изд. А. С. Суворина, 1907, с. 573.

[27] Дроздов Ю. И. Вымысел исключен (записки начальника нелегальной разведки). Альманах «Вымпел», М., 1997, с. 21.

[28] РГАДА. Ф. 1261. Оп. 3. Д. № 1129. «Письма Гольдбаха вице-канцлеру [М. И. Воронцову]». Л. 1–1об.

[29] РГАДА. Ф. 199. Оп. 1. Ед. хранения № 247. Д. № 15.

[30] СПбФ АРАН. Ф. 136. Оп. 2. Д. 5. Л. 19, 19об. Письмо Эпинуса к Л. Эйлеру от 8 июня 1757 г.

[31] Jacobi Bernoulli. Profess. Basil. & utriusque Societ. Reg. Scientiar. Gall. & Pruss. Sodal. Mathematici Celeberrimi, ARS CONJECTANDI, opus posthumum. Accedie Tractatatus de Seriebus Infinitis, et Epistola Gallicéscrita De Ludo Pilæ reticularis. Basileæ, Impensis Thurnisiorum, Fratrum. clɔ lɔcc xiii.

[32] СПбФ АРАН. Разряд V. Оп. Э-7. Ед. хр. 32. Л. 1–2об.

[33] Leighton A. C. Some examples of historical cryptanalysis. Historia Mathematica. 1977, v. 4, № 3, p. 319–337.

[34] АВПРИ. Ф. «Шифровальный отдел». Оп. 480/3. Д. № 5258. Л. 11–19об.

[35] АВПРИ. Ф. «Секретные мнения». Д. № 591, часть I. Л. 19.

[36] АВПРИ. Ф. «Секретные мнения». Д. № 591, часть I. Л. 18об.

[37] РГАДА. Ф. 10. Оп. 1. Д. № 434. Л. 1, 2.

[38] Ларионова Т. С. Немецкие издатели в России (XVIII–начало XX вв.). В сб.: Немцы в России. Три века научного сотрудничества. СПб.: изд. «Дмитрий Буланин», 2003, с. 370.

[39] Архив князя Воронцова. Книга 9. М.: 1876, с. 408, 409.

Часть II
Обзорные доклады

Проблемы математического, алгоритмического и программного обеспечения компьютерной безопасности в Интернет

В. А. Васенин

1 Введение

Обеспечение информационной безопасности (ИБ) в Интернет — сфера деятельности, требующая для своей реализации решения целого ряда сложных задач. Она не только многопланова по числу направлений административно-организационной и научной-технической активности, но и по количеству уровней иерархии, на которых она организована. Мировая практика показывает, что к обеспечению безопасности национально-значимых информационных систем в Интернет и средств телекоммуникаций привлекаются администраторы от руководителей государственного ранга до специалистов, ответственных за сопровождение отдельных функциональных сервисов. Разработкой и реализацией решений на отдельных уровнях обеспечения ИБ занимаются ученые гуманитарного цикла (юристы и специалисты по управлению, психологи и социологи), исследователи-естественники — математики и физики, инженеры-программисты и электронщики. Нельзя, например, построить систему ИБ крупной корпорации без создания с высокой гарантией защищенности отдельных составляющих её информационных ресурсов, объединяющих их сетевых узлов или поддерживающих функциональных сервисов (e-mail, Web и т. п.) Изложенные доводы — свидетельство комплексного характера проблемы.

Не умаляя важности других на широком спектре проблем и отдельных задач ИБ, в данной публикации остановимся на тех, которые связаны с их математическим, алгоритмическим и программным обеспечением. Данную работу следует рассматривать как попытку изложить точку зрения на состояние и перспективы этого направления в связи с решением проблем ИБ в Интернет.

На всех уровнях создания системы информационной безопасности любого продукта, системы информационных технологий, начиная с административного и выработки политики безопасности до программно-технического, обеспечивающего защиту (реализующего функции, направленные на защиту) информации в отдельной («монополитной») компьютерной системе или на сетевой среде от внешних или внутренних угроз математическое, алгоритмическое и программное обеспечение играет решающую роль. Там, где удается построить математические модели защиты или использовать строгие правила критериального подхода, закладывается фундамент доказательной базы, надежно гарантирующей уровень защищенности проектируемого, аппаратно и программно реализуемого, а затем и эксплуатируемого объекта (продукта или системы) информационных технологий (ИТ), будь то операционная систем, база данных или сетевая инфраструктура управляющего узла корпоративной сети.

Остановимся на отдельных аспектах этого направления, важность которого обусловлена появлением на Интернет большого числа информационно-вычислительных систем, активно влияющих на многие сферы хозяйственной деятельности, в том числе — стратегически важные для государства.

2 Формирование политики безопасности

Одним из ключевых элементов уже на этапе проектирования любого объекта[1] — продукта или системы информационных технологий (в нотации, принятой, например, в общеизвестных критериях [1, 2, 3, 4]), который затем трансформируется в требования к моделям и сценариям его защиты, а также сервисам и механизмам их реализации в аппаратуре или программном обеспечении, является *политика безопасности* этого объекта. Здесь и далее под политикой безопасности (ПБ) будем

[1] Здесь и далее будем употреблять понятие объект ИТ (объект), имея в виду продукт или систему ИТ, отличая его от объекта традиционной «субъектно-объектной» модели, описывающей схемы разграничения доступа.

понимать только аспекты, связанные с политикой информационной безопасности или защитой информационных ресурсов и поддерживающей их сетевой инфраструктуры, принимая справедливость тезиса «компьютер — это сеть», впрочем, также как и тезиса «сеть — это компьютер».

Формирование ПБ объекта происходит поэтапно, как минимум на двух взаимообуславливающих друг друга уровнях. На верхнем уровне она связана положениями, затрагивающими отношение к объекту организации, ведомства или корпорации, которым этот объект принадлежит (которая его разрабатывает или эксплуатирует). Эти положения носят общий характер, отражая цели деятельности, отношение к потенциальным угрозам, включают анализ рисков, способы и формы работы с информацией, административные акты и программу ИБ, регламентирующие деятельность персонала на этом направлении, а также ряд других аспектов.

На нижнем уровне формируется собственно ПБ объекта, как набор норм, правил и практических приемов, которые регулируют управление объектом (информацией), обеспечивая его защиту от потенциальных (внешних, внутренних, злоумышленных или других) угроз. Это может быть, например, набор правил управления доступом к объекту, практическая реализация которых осуществляется как на операционном (например, отдельные комнаты со средствами доступа к сетевым ресурсам и прямым контролем за действиями пользователя) или программно-техническом (с учетом идентификации и аутентификации пользователя, авторизации, квотирования ресурсов и т. п.) уровне. Управление доступом к ресурсам современных систем должно эффективно сочетать преимущества произвольного (на основе учета личности субъекта или группы) и принудительного (на основе меток безопасности) начал, а также их совместимость с элементами обеспечения безопасности повторного использования объектов.

Следует отметить, что на практике, особенно для стратегически важных современных объектов, формирование ПБ — сложный, многофакторный процесс. С появлением глобального транснационального информационного пространства на базе Интернет(иногда именуемого киберпространством) эта, без того непростая задача, стала намного сложнее. Сегодня ошибки в формулировании ПБ объекта оборачиваются созданием «дорогостоящей» (ресурсоемкой по затратам) системы, которая на практике не способна в полной мере выполнять своих функций. В более «жесткой» постановке [5] системы, обеспечивающие стратегически важные функции по поддержанию системообразующих сфер национальной экономики (например, авиационный или железнодорожный транспорт, энергетический или сырьевой комплексы) могут оказаться под угрозой.

Анализ положений доктрины информационной безопасности (ИБ) России [6] позволяет утверждать, что создание целостной системы ИБ (в первую очередь, — государственной) на российском сегменте Интернет является одной из самых неотложных задач. Она очень трудна для решения в связи с масштабами государства, многоукладностью экономики и объективно существующей в этой связи спецификой требований отдельных составляющих ее отраслей, интересов разных хозяйствующих субъектов, корпораций и ведомств. Эффективное решение задачи национального масштаба возможно только на пути формирования подходов к созданию систем ИБ на основе ПБ, учитывающих эту специфику. Например, и это очевидно, что положения ПБ объектов, принадлежащих организациям из так называемых «силовых структур» должны отличаться от соответствующих положений для объектов корпораций, работающих в банковской сфере или занимающихся добычей и сбытом природных ресурсов. Однако и для организаций упомянутых сфер деятельности, также как и для других, которые объединяет Интернет, разработка политики безопасности очень важная задача. Нельзя эффективно защищать свое в «общем котле», не обозначив заранее принципов и правил, на которых такая защита осуществляется. В этом проявление специфики Интернет, как объединения сетей разных субъектов-собственников, уровней и протокольной базы, из разных регионов и даже стран. Таким образом, становится очевидным, что формирование ПБ для ведомств и корпораций, занимающихся родственной деятельностью была бы первым эффективным шагом в направлении создания целостной системы ИБ на российском сегменте Интернет.

Методические основы (предпосылки) в виде критериальной базы для практических действий на этом направлении существуют. К их числу можно отнести, хотя и не в полной мере отвечающие потребностям сегодняшнего дня «Руководящие документы Гостехкомиссии при Президенте РФ по защите от несанкционированного доступа» [7, 8, 9, 10, 11], положения других зарубежных Критериев, в том числе «Общих Критериев» (ISO 15408) [1, 2, 3, 4].

Детальный анализ и формально строгий учет факторов (структура ценностей и уровни доступа, анализ рисков, каналы утечки информации и пр.), влияющих на информационную безопасность объекта на этапе формулирования или синтеза для него политики безопасности, а затем на этапе анализа

эффективности ее реализации, оценки уровня гарантированной защищенности — важнейшие задачи. Успешное и строгое решение этих задач возможно только на основе формализации и переноса основных положений ПБ на математические модели. В соответствии с уровнем сложности объекта, для которого разрабатывается ПБ, возможна его декомпозиция на отдельные составляющие и, соответственно, построение системы взаимосвязанных моделей. Такими, например, могут быть модели ИБ для операционных систем, используемых при построении разных сегментов крупных корпоративного масштаба информационных комплексов, а также модели ИБ отдельных объектов — подсистем (БД, серверов, обеспечивающих функциональные сервисы или Web-ресурсы). В качестве составляющих могут рассматриваться модели ПБ управляющих сегментов отдельных сетей в составе крупной корпоративной сети. В общей постановке это очень сложная задача и эффективного ее решения пока не найдено.

Если рассматривать ПБ с точки зрения набора правил управления доступом к ресурсам компьютерных систем (см., например, [12]), то сегодня, как правило, применяется модель дискреционного доступа, механизмы реализации которой имеются почти во всех операционных системах (ОС) и позволяют оперативно и гибко настраиваться на требуемую функциональность. Однако эта модель не имеет строгой доказательной базы, гарантирующей защищенность системы, не обеспечивает контроля распространения прав доступа и плохо защищает от скрытых средств разрушающего воздействия («троянские кони», вирусы). В последние годы все большее распространение стала получать лишенная этих недостатков, менее вариативная функционально, но более жесткая в выполнении правил разграничения доступа многоуровневая модель ПБ, которая реализуется через мандатный, принудительный контроль доступа, основанный на решетке ценностей (соотношение ценностей объектов и уровней доступа отдельных субъектов). Ориентированные изначально на сохранение секретности информации положения этой модели, модифицированные с учетом предложений Biba [13], формируют модель, способную с успехом решать проблемы защиты от угроз нарушения целостности.

В настоящее время совершенствуется и находит все большее практическое применение ролевая [12] (адаптирующаяся к изменениям в полномочиях доступа к информации), а также ряд других моделей, использующих положительные и оправдавшие себя на практике подходы. Ролевой подход, сочетающий преимущества дискреционного и мандатного принципов разграничения доступа, создает предпосылки и закладывает механизмы для расширения возможностей традиционных моделей применительно к современным системам, с учетом новых степеней свободы и дополнительных факторов, привносимых распределенной сетевой средой и Интернет.

При всем многообразии существующих традиционных подходов к построению моделей ИБ, использование их для составления моделей информационной безопасности больших распределенных в Интернет систем связано с необходимостью учитывать ряд дополнительных факторов. К их числу относятся:

- изменяющееся (иногда перманентно, как в системах типа GRID) количество составляющих компонент, и, как следствие, во много раз возрастающее число возможных состояний системы;
- сложности фиксации периметра безопасности даже для корпоративных систем, не говоря уже о системах, имеющих выход в Интернет (что необходимо по объективным причинам);
- неподконтрольность состояния каналов связи между компонентами системы, высокий уровень опасности утечки информации, в том числе через скрытые каналы и т. п.;
- объективно ограниченные скорости обмена данными между компонентами, что создает сложности для организации монитора обращений для системы в целом;
- к числу традиционных угроз секретности и целостности информации добавляется менее традиционная для «моносистем» угроза доступности — потеря возможности авторизованному пользователю получить доступ к информации.

Строгая формализация ПБ для больших распределенных систем с учетом изложенных факторов — очень трудная задача. Унифицированного, а тем более, эффективного с практической точки зрения подхода к ее решению пока не предложено. Именно это обстоятельство главным образом стимулировало появление и поддерживает развитие критериального подхода к оценке степени надежности средств защиты сложных объектов. Однако, и это хотелось бы подчеркнуть, возможности математически строгих подходов к построению моделей реализации ПБ для распределенных систем в Интернет

далеко не исчерпаны. Новые идеи должны быть основаны на более глубоком осмыслении изложенных выше факторов, характеризующих распределенные системы в Интернет, подходов к составлению политик их безопасности, которые прошли теоретическую или практическую апробацию. Необходимо учитывать разные требования к использованию и, соответственно, защите ресурсов в различных сегментах сети, например, ее управляющего ядра, где обеспечиваются основные инфраструктурные и функциональные сервисы, реализуются технологии среднего уровня (middleware) и периферийных сегментов, где, как правило, размещаются прикладные сервисы. Разнятся требования к использованию ресурсов различных сегментов, так называемых «виртуальных организаций» в системах типа GRID, где высший межсегментный уровень, аккумулирующий данные о всех ресурсах системы и механизмы работы с ними, должен быть организован и защищен иначе, чем нижележащий уровень линейных сегментов. В основу политики безопасности, учитывающей отмеченные особенности использования ресурсов разных сегментов сети может быть положен, например, подход, основанный на так называемой зональной модели разграничения доступа [15]. Подобный подход способен обеспечить сочетание (совмещение преимуществ) традиционных дискреционной и мандатной модели разграничения доступа на уровне отдельных сегментов (зон) с межсегментной (общей, или межзональной) политикой.

Конечно, отмеченный подход лишь один из возможных. С учетом изложенных обстоятельств необходим поиск новых способов к формализации ПБ. К их числу можно отнести следующие.

- Разработку новых или эффективное использование уже существующих языков моделирования, позволяющих на концептуальном уровне построения модели ПБ отобразить семантику проблемной области. Такие модельно-языковые средства способны обеспечить более адекватное (полное и математически строгое) описание моделей систем, реализующих заданную ПБ и, как следствие, доказательную базу ее гарантированной защищенности.

- Создание моделей ПБ, учитывающих специфику распределенных систем на гетерогенной среде Интернет, использующих в качестве базовых графовые и автоматные, статистические, детерминированные и другие, как традиционные так и нетрадиционные способы их формализации. Учет факторов, характеризующих описанные выше особенности больших распределенных систем, будет также способствовать совершенствованию доказательной базы их гарантированной защищенности.

- Разработка подходов к построению моделей ПБ на основе совершенствования традиционных — произвольного (дискреционная модель), принудительного (мандатный контроль), ролевого доступа и их комбинации для различных структурных элементов (компонентов), сервисов и приложений больших распределенных систем. Рациональное использование различных моделей в составе больших систем способно повысить уровень их защищенности, сократить время и обеспечить экономию вычислительных ресурсов на реализацию мер, предусмотренных ПБ.

Вместе с тем, объективный анализ положения дел показывает, что практических шагов на этом направлении в России пока очень мало. Об этом свидетельствуют, например, результаты и оценки, приведенные в [16]. К причинам такого положения дел, в первую очередь, следует отнести отсутствие:

- должной популяризации задачи, подходов и способов ее решения среди широких слоев населения, действий просветительского и образовательного плана;

- стимулов со стороны государственных структур и побудительных мотивов для соответствующих ведомств и корпораций к практическим действиям на этом направлении.

Некоторые действия на этих направлениях все-таки предпринимаются. С позиций популяризации и просветительства следует отметить несколько книг российских авторов, освещающих проблемы и подходы к решению задач на этом направлении. К числу наиболее значимых можно отнести «Теоретические основы защиты информации» А. А. Грушо и Е. Е. Тимониной (1996 г.) [12], «Информационная безопасность. Практический подход» (1998 г.) [17], «Основы информационной безопасности» В. А. Галатенко (2003 г.) [18], «Основы безопасности информационных систем» П. Д. Зегжды и А. М. Ивашко (2000 г.) [19], «Введение в теорию и практику компьютерной безопасности» А. Ю. Щербакова (2001 г.) [20]. Все книги анализируют положение дел в области ИБ на основе систематизации подходов к построению защищенных объектов с позиций, изложенных в зарубежных критериях оценки надежных систем и Руководства Гостехкомиссии РФ. Рассматриваются проблемы, связанные

с построением строгих моделей ИБ, математических методов и оценки гарантированной защищенности систем. Есть и другие публикации на эту тему, в том числе в виде книг по смежной тематике, научных и научно-популярных статей, информационного сайта «cryptography.ru», который также затрагивает эти проблемы. Однако практика показывает, что этого мало для того, чтобы сдвинуть дело с «мертвой точки».

Рассматривая образовательную составляющую проблемы в целом, следует отметить, что в вузах России (гражданского профиля), особенно в последние годы (с выходом в свет Доктрины ИБ России) [6] и усилением внимания к этой проблеме со стороны государственных структур (Совета Безопасности РФ — в первую очередь), появились лаборатории, кафедры и даже факультеты ИБ, не говоря уже о специальных курсах в учебных планах. Однако, эти кафедры, как правило, достаточно узко специализированы на отдельные направления ИБ, ориентированы на проблемы «заказчиков» (банковское дело, потребности «силовых ведомств» и т. п.). Знания, получаемые в аудиториях, не подкрепляются практикой, а тем более исследованиями на сетевых полигонах с использованием современных технологий. У выпускников не формируется «широкий» взгляд на проблему и пути ее разрешения. Такое положение дел не способствует формированию слоя специалистов-исследователей в области ИБ, способных активно распространять свое влияние на деятельность корпораций, ведомств, формировать общественное мнение, способствуя тем самым решению задач в свете положение Доктрины ИБ России.

3 Доказательная база надежности реализации политики безопасности

После того, как ПБ сформулирована и принята, способы ее реализации представляются определенным набором методов, правил и механизмов, интегрированных в модели и сценарии, которые призваны гарантировать соблюдение политики. Такой набор реализуется в виде аппаратного и программного комплекса — самостоятельной подсистемы или продукта информационных технологий. Любой из подобных комплексов требует оценки его возможностей в полном объеме реализовать все положения ПБ. Гарантированность, как мера уверенности в том, что инструментальные средства защиты обеспечивают выполнение принятой ПБ, главным образом подтверждается на архитектурном и технологическом уровнях их реализации.

С позиции архитектурного уровня особая роль принадлежит защите ядра ОС, четкой систематизации привилегий для выполнения аппаратных и системных функций, их минимизации для отдельных компонент, сегментации адресного пространства процессов и т. п. Эти положения в той или иной мере будут затронуты в разделе 4, а также при обсуждении проблем, связанных с совершенствованием сервисов программно-технического уровня.

С точки зрения гарантированности технологической можно выделить следующие подходы, требующие для своего осуществления математических моделей, алгоритмического и программного обеспечения:

- тестирование;
- верификация на основе формальных методов доказательства;
- математические модели гарантированно защищенных систем.

Тестирование — традиционный способ, сопровождающий приемо-сдаточные испытания любых, в том числе компьютерных систем ИБ. Сложность задачи на сегодня состоит в создании моделей и алгоритмов, предъявляющих адекватные требования к современным системам, объединяющим, как правило, ресурсы на гетерогенной сетевой среде. Такие требования к инструментальным средствам защиты должны предъявляться не только на начальном этапе (разработка и приемо-сдаточные испытания), но и во время эксплуатации, в условиях объективного изменения ее состава и функциональности. Методической основой для действий на данном направлении являются положения критериальных подходов, изложенные в соответствующих, как отечественных, так и зарубежных документах, например [1, 2, 3, 4].

Верификация в автоматизированном режиме на основе формальных методов доказательства — эффективный подход к обоснованию гарантированной защищенности компьютерных систем на всех

этапах их жизненного цикла. Одним из важных является направление, связанное с созданием таких систем для обеспечения безопасности сложных программных комплексов. Актуальность данного направления повышается в связи с повсеместным использованием в составе таких комплексов программных средств, которые свободно распространяются через Интернет. Хорошие перспективы имеет использование систем автоматической верификации состояния защищенности в процессе функционирования практически значимых распределенных систем. Однако, несмотря на определенные теоретические заделы и отдельные прикладные разработки, эффективных решений для подобных систем пока нет. Такое положение, в первую очередь, связано с объективной трудностью их формального описания, учета многофакторных процессов, которые следует принимать во внимание. Необходим поиск подходов к созданию подсистем, автоматизирующих процессы верификации безопасности отдельных функционально замкнутых компонентов, описанию и практической реализации моделей их взаимодействия в составе больших распределенных систем. Теоретические основы для решения подобных задач, подходы к формализации программ и потенциальных средств разрушающего воздействия, методы анализа и инструментальные средства на этом направлении только создаются.

Если модель ПБ удается достаточно подробно и четко формализовать математически, то высока вероятность получить аналитически (привлекая математический аппарат) или путем строгих логических рассуждений доказательство гарантии выполнения объектом защиты ПБ. Для относительно простых систем и перечисленных выше моделей ПБ в условиях достаточно «жестких» дополнительных ограничений получить такие оценки можно. Их удается получить, например, в рамках модели «Take-Grant» [12], описывающей с помощью теории графов способы распространения прав доступа в системах с дискреционной ПБ, или теоретико-множественных моделей «Белла — Лападула» [21, 22], «Low-Water-Mark» [23, 24], автоматной модели невлияния «Гогена — Месгауэра» [25, 26] для многоуровневых систем (мандатный контроль доступа), а также ряда других.

Однако практические системы, подлежащие защите, являются более сложными и, как правило, распределенными. Создание модели системы в виде набора математически строгих соотношений, описывающих ее начальное состояние и ограничений на возможные переходы (во времени и фазовом пространстве), не нарушающие ПБ многопараметрической (многокомпонентной) системы, которые позволяли бы путем столь же строгих логических рассуждений доказать справедливость выполнения ПБ в любых состояниях системы, оказывается очень трудной задачей. С учетом объективно высокого уровня сложности подобных систем и огромного числа возможных ее состояний в процессе эксплуатации требуются новые нетрадиционные подходы к построению математических моделей гарантированно защищенных распределенных объектов ИТ. В качестве примера, подтверждающего перспективность такого подхода можно привести модель распределенной компьютерной системы, представленной в виде обобщения автоматной модели невлияния Гогена — Месгауэра на случай вероятностного автомата [27]. Такой подход позволил редуцировать традиционную детерминированную модель с огромным числом состояний к вероятностной модели с таким их числом, которое позволяет построить достаточные и почти необходимые локальные условия, обеспечивающие глобальную безопасность системы. Эффективность новых подходов к созданию математических моделей систем, гарантированно защищающих распределенную систему от атак на доступность, продемонстрирована тем же автором в работе [28].

Важным направлением исследований является поиск способов гарантированной защиты данных в недоверенной среде на основе применения надежных алгоритмов шифрования, применяемых, например, в межсетевых шлюзах. Отдельно в ряду задач на этом направлении стоит поиск механизмов и выработка моделей организации (скрытых) каналов утечки информации, включая способы управления ими, а также выработка и реализация мер противодействия. Результаты на этом направлении изложены в работах [12, 14].

Политика безопасности, модели, механизмы и средства защиты многих из практически значимых систем представляют государственную тайну и являются «закрытыми». Эти обстоятельства заставляют искать другие подходы к доказательству гарантированности (высокой степени доверия) для объектов защиты с точки зрения выполнения ими положений принятой ПБ. Гарантированность в данном случае обеспечивается на операционном, архитектурном, как одном из его аспектов, и технологическом уровнях реализации инструментальных средств защиты объекта.

Таким является подход, основанный на формулировании условий в виде стандарта, с высокой степенью вероятности гарантирующего поддержку ПБ. Первым подобным стандартом стала «Оранжевая книга» [29] (США, впервые опубликована в 1983 г.), которая гарантировала поддержку ПБ, основанную на дискреционной и мандатной моделях, для изолированных «монолитных» на отдель-

ных компьютерах информационных систем, затем дополнения к этому стандарту, гарантирующие защиту распределенных систем (1987 г.) (Интерпретация «Оранжевой книги» для сетевых конфигураций [30, 31, 32] и распределенных баз данных [33] (1991 г.) (Рекомендации X.800)) на основе тех же политик. Подход оказался продуктивным и стимулировал разработку как ряда национальных, так и международных стандартов [34, 35, 36]. Последним из них стал результат 5-летней работы специалистов из США, Канады, Англии, Франции — стандарт ISO/МЭК 15408 «Критерии оценки безопасности информационных технологий», опубликованный в 1999 г. [1, 2, 3] и более известный как «Общие критерии». «Общие критерии» объединили в себе все лучшее критериальных подходов прошлых лет и в настоящее время стал «дефакто» мировым эталоном в этой области.

Проблемы разработки и строгой формализации ПБ, доказательства гарантий ее выполнения для практически значимых, важных объектов (продуктов и систем ИТ) в последние годы в значительной степени связаны с их сложной внутренней структурой и принципами организации внешней сетевой среды функционирования, в качестве которой, как правило, выступают сети Интернет [37]. К числу таких проблем можно отнести:

- независимость и равноправие сетей, объединенных в Интернет на основе стека протоколов TCP/IP, и проблемы ограничений сетевого доступа без ущерба для узла, инициирующего такие ограничения;
- межведомственный, региональный и транснациональный характер сети, сложности применения на ней законодательных мер, административных рычагов и операционных регуляторов;
- практическое отсутствие возможностей защиты каналов связи для крупных сетей пакетной коммутации и объективные сложности создания модели единой системы, гарантирующей защиту взаимодействующих объектов на таких сетях;
- сложности решения проблемы аутентификации пользователя (субъекта) в условиях удаленного сетевого доступа без применения идентификаторов, учитывающих его индивидуальные особенности;
- высокие темпы изменения (мобильности) субъектов доступа и объектов защиты на сетях, а также связанные с этим сложности описания их ПБ.

Отдельные задачи по каждой из этих проблем, а также в их взаимосвязи решаются в рамках приоритетных и долговременных программ. Такие программы реализуются как в отдельных странах, так и на уровне международной кооперации. В последние годы главные результаты на этом направлении получены на основе критериальной базы, позволяющей разрабатывать и реализовывать ПБ объектов ИТ на основе функциональных требований к элементам системы защиты, задания по безопасности объекта и профиля его защиты. Российскими специалистами на основе Общих критериев разработаны предложения по внедрению в стране государственного стандарта ГОСТ Р ИСО/МЭК 15408 «Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий» [4, 38]. В настоящее время идет активное обсуждение способов его внедрения. Это также сложный и долговременный процесс. Исследования гарантий защищенности объектов на основе критериальной базы — это важное и эффективное направление, однако возможности строгих математических моделей ПБ и аналитической доказательной базы эффективности ее реализации далеко не исчерпаны. Это относится как к «монообъектам», так и к продуктам ИТ, представляющим собой распределенные сетевые системы на метасети Интернет.

4 Операционная система — главный объект защиты и исходный рубеж информационной безопасности

Примером «монолитного» объекта, который требует особого внимания, с точки зрения информационной безопасности, является операционная система (ОС). ОС — набор программных компонент (в том числе — микропрограммных), которые обеспечивают поддержку аппаратуры компьютера с целью управления его ресурсами, включая процессоры, память, устройства ввода-вывода, данные. Степень защищенности любых, в том числе сложных, распределенных систем в Интернет, в большей степени

зависит от механизмов реализации инфраструктурных сервисов, которые, как правило, поддерживаются ядром ОС.

Инициаторами запросов к ОС выступают как пользователи (в том числе, — операторы, программисты или администраторы), так и программы, аппаратные средства. С этих позиций ОС сама представляет собой комплекс, объекты информации и субъекты-потребители которого взаимодействуют между собой с использованием тех же первичных механизмов (механизмов ядра ОС), что и объекты и субъекты отдельного «монолитного» компьютера или компьютеров, распределенных на сети. Более того, совокупность таких механизмов обеспечивает поддержку базовых программно-технических сервисов безопасности в ядрах всех современных ОС (включая идентификацию/аутентификацию, разграничение доступа, протоколирование и аудит). С этих позиций все проблемы вышеперечисленных программно-технических сервисов ИБ, которые будут осуждаться далее в разделе 5, в той или иной мере относятся и к ОС, во-первых, как к продукту ИТ, во-вторых, как к месту их размещения и эффективной поддержки. Отмеченные особенности ОС указывают на то, что ее надежная защита — главная задача и начальная гарантия уровня защищенности информационно-вычислительного объекта любой, сколь угодно сложной архитектуры и назначения.

С учетом особой роли операционной системы поиск подходов к устранению «уязвимостей» как «локально-тактического», так и «принципиально-стратегического» характера, которые объективно существуют на сегодня в ОС для любых аппаратных платформ, является важнейшим направлением [39]. По-прежнему остаются проблемы защиты от скрытых средств разрушающего воздействия и приобретения неадекватных полномочий пользователем при реализации ПБ на основе традиционной дискреционной модели доступа. Механизмы принудительного доступа (на основе многоуровневой модели), в большей степени защищающие от подобных атак, имеют свои сложности реализации в ядрах традиционных ОС и проблемы практического использования при создании монитора безопасности для больших распределенных систем.

К уже упомянутым при использовании компьютерных систем в Интернет добавляются угрозы отказа в обслуживании, которые, как правило, реализуются путем изменения настроек подсистемы сетевого доступа. Однако следует отметить, что возможности традиционных как дискреционных, так и многоуровневых моделей, реализующих произвольный и принудительный доступ, на данном направлении далеко не исчерпаны. Для создания дополнительных, более тонких механизмов защиты от рассмотренных выше угроз необходимо выделить в отдельную группу объектов сущности ядра ОС, которые отвечают за доступ к сети. Такими могут быть, например, локальные адреса используемых протоколов, порты, записи в таблицах маршрутизации, правила пакетных фильтров [40, 41]. За доступом к объектам данной группы должен быть обеспечен отдельный контроль. На этом пути необходимо реализовать принцип наименьших привилегий по отношению к типовым сетевым задачам (сервисам). С этой целью кроме традиционно используемых идентификаторов пользователя и процесса представляется целесообразным ввести дополнительные идентификации подобных задач (сервисов). Тогда полномочия на использование средств сетевого доступа (управления сокетами, конфигурацией, создание фильтров и т. п.) можно будет определять с учетом приоритетов задачи, которая эти средства использует.

Эффективная реализация указанных мер связана с необходимостью анализа проблем на данном направлении, исследованием и систематизацией уже имеющихся подходов к их разрешению и разработкой достаточно строгой, поддающейся верификации и тестированию модели системы защиты ОС. Подобные исследования на сегодня активно проводятся за рубежом [42, 43, 44]. Есть работы на этом направлении и в России.

В разделах 2 и 3 уже отмечалась важность более широкого внедрения перспективных с точки зрения современных требований политик безопасности, в первую очередь — многоуровневой, реализующей принудительное управление, а также ролевой и других. Необходимость таких действий диктуется требованиями современных критериальных подходов к условиям эксплуатации практически значимых систем. Наряду с традиционными сервисами (идентификации и аутентификации, протоколирования и аудита) для реализации подобных дополнительных средств защиты необходим поиск новых эффективных моделей, программных решений в ядре ОС, не затрагивающих других, уже сложившихся и широко используемых механизмов. Такие задачи, особенно в последние годы, решаются в России. На сегодня есть ряд ОС, реализующих многоуровневую ПБ на основе мандатного контроля и прошедших различные уровни апробации, например [45, 46]. Есть разработки, коды которых представлены в открытом доступе.

Одной из важнейших остается задача создания отечественной ОС, отвечающей предъявляемым

к ней современным требованиям по функциональности. С учетом объективных сложностей на пути решения этой задачи в полном объеме, на начальном этапе необходим поиск подходов, отвечающих хотя бы минимальному набору таких требований, которые диктуют национально значимые системы и (или) объекты высокой практической ценности.

С этих позиций следует рассматривать разработку НИИСИ РАН операционной системы ос 2000 [47], оригинальные архитектурные решения которой, кроме удовлетворения ряда специальных функциональных требований, призвана обеспечить дополнительные возможности собственной защиты и безопасности прикладных систем на ее базе. Требования к функциональности ОС и оценочные уровни доверия к ней сформулированы в соответствии с положениями «Общих критериев» [1, 2, 3].

К значимым разработкам на данном направлении следует отнести создание POSIX-совместимой [48] защищенной ОС «Феникс» в Специализированном центре защиты информации СПбГТУ. В основу архитектурных требований к данной системе положен оригинальный подход к управлению доступом, трактовка защищаемых сущностей в виде информационных ресурсов и универсальный интерфейс доступа.

Подчеркивая значимость указанных выше результатов в направлении создания новых и совершенствования уже существующих ОС, следует отметить, что сегодня ни одна из них не ориентирована должным образом на использование в составе распределенных систем на существенно гетерогенной инфраструктуре Интернет. Создание таких ОС — дело будущего. В настоящее время необходимо более точно сформулировать перечень требований к такого сорта системам, разработать модель, которая, в свою очередь, должна обеспечить необходимую функциональность системы и требуемый уровень доверия к ней.

5 Сервисы информационной безопасности программно-технического уровня

Анализируя обсуждаемые выше модели и сценарии защиты объекта, реализующие ПБ на программно-техническом уровне, необходимо отметить, что основными составляющими (компонентами, элементами) интегрированного набора средств защиты являются сервисы ИБ. Некоторые из них, как правило, проблемно-ориентированные (например, защита отдельных сервисов, высокопроизводительных вычислительных систем, крупных коммуникационных узлов сетей, банковских документов и т. п.) или функционально ориентированные (экранирование, идентификация-аутентификация, управление доступом и т. п.) являются составными и включают в себя более мелкие, в том числе атомарные (или неделимые) сервисы.

К числу базовых среди сервисов ИБ можно отнести сервисы, выполняющие следующие функции:

- защита среды передачи;
- защита данных при передаче;
- экранирование и фильтрация (защита периметра);
- идентификация / аутентификация;
- разграничение доступа;
- протоколирование и аудит.

Отдельные из перечисленных сервисов или их комбинации составляют ядро модели (сценария), реализующей ПБ любой, сколь угодно сложной системы информационных технологий, включая комплекс, объединяющий распределенные сетевые объекты.

Кратко остановимся на особенностях каждого из функциональных сервисов с точки зрения задач, связанных с развитием их математического, алгоритмического и программного обеспечения.

Защита среды передачи

Рассматривая защиту среды передачи (или физическую защиту коммуникаций) следует иметь в виду, что традиционно она реализуется с помощью:

- защиты поддерживающей инфраструктуры;
- физического управления доступом;
- противопожарных и других подобных мер.

Приведенный перечень содержит действия, которые принято относить к мерам операционного уровня иерархической схемы ИБ. Однако, в условиях больших распределенных систем в Интернет, эти меры в большей степени должны быть ориентированы не на людей (персонал), а на программно-технические средства.

Каждое из действий в рамках перечисленных мер не требует разработки математических моделей, алгоритмизации и (или) реализации сложного программного обеспечения для эффективного выполнения предусмотренных функций. Контроль за состоянием зданий и сооружений, отдельных помещений (в т. ч. управляющих коммуникационных узлов сети), систем электроснабжения и кондиционирования, состояния противопожарных средств традиционно связаны с рычагами (мерами) вышележащего (по отношению к программно-техническому) операционного уровня реализации ПБ (ИБ). Ключевым звеном, через которое эти меры осуществляются, является человек (специалист). Защита от перехвата данных также реализуется персоналом с помощью мер физического контроля несанкционированного доступа (использования) линий связи или с помощью специально спроектированных приборов, основанных на физических эффектах (принципах).

Однако, если рассмотреть данный сервис в совокупности перечисленных мер для большой распределенной системы, штат персонала, который призван выполнять отдельные действия (функции) по контролю состояния всех средств защиты среды передачи и своевременного реагирования на нештатные ситуации, то возникает настоятельная и объективная потребность в создании комплекса, обеспечивающего эффективный мониторинг состояния столь сложной системы и выработку рекомендаций для оперативного управления ею. С учетом многообразия технических средств контроля, которые используются на каждом из направлений, существенно территориально-распределенного характера их размещения в качестве важной и очень перспективной должна рассматриваться задача создания подсистемы мониторинга за состоянием технических средств, обслуживающих защиту среды передачи. Такая подсистема в общей системе управления сетью может быть построена, например, для управляющего сегмента сети отдельной организации или для магистральной инфраструктуры корпоративной сети.

Задача в такой постановке требует:

- учета существенно распределенного характера агентов — технических средств поставщиков и (или) потребителей информации;
- анализа типовых бизнес-процессов, протекающих в системе и обеспечивающих функциональное взаимодействие данных, собираемых с различных устройств.

Перечисленным требованиям отвечает инструментальный комплекс, поддерживающий взаимодействие распределенных агентов, выступающих с одной стороны, как объекти-сточник, а с другой, — как субъект-потребитель информации о состоянии безопасности среды передачи. Структура организации такой подсистемы должна быть иерархической, а разработку ее программного обеспечения целесообразно осуществлять на принципах объектно-ориентированного подхода. Архитектура и технологические принципы этого комплекса (подсистемы) должны быть рационально интегрированы в общую систему управления сетью. Создание подобной системы безусловно потребует разработки новых, адекватных математических моделей, сложных алгоритмов и соответствующего программного обеспечения. Как описание общих подходов к созданию подобного сорта и уровня сложности систем можно рассматривать работы [49, 50].

Защита данных

Защита данных при передаче по каналам связи осуществляется с помощью средств криптографии. На этом направлении существует много результатов, основанных на самых современных математических методах. Эти работы, как правило, носили «закрытый» характер и, как следствие, большинство результатов становилось доступно широкой общественности с большим опозданием. Причиной тому является их стратегическое значение и постоянное внимание к этим проблемам со стороны оборонных

ведомств и силовых структур любого государства. В последние годы, в связи с развитием сетевых инфраструктур на национальном и межнациональном уровне, активным использованием сетевых технологий в других сферах хозяйственной деятельности, в бизнесе и медицине, сфере развлечений и на бытовом уровне, проблемы конфиденциальности информации стали интересны для более широкой аудитории. Как следствие, многие из подходов, проблем и задач на этом направлении, решения, стандарты и протоколы становятся достоянием широкой общественности. К числу таких работ российских авторов следует отнести [51], материалы конференции «Московский университет и развитие криптографии в России» (октябрь 2002 г.) [52].

Хотелось бы отметить особенности, которые накладывает Интернет на криптографию, как вид сервиса безопасности. Одним из основополагающих принципов Метасети является его открытая архитектура, методологическую основу которой составляет модель открытых систем [53]. Именно этот принцип в сочетании с масштабируемостью и простотой реализации базовых протоколов обеспечивают Интернет беспрецедентно высокие темпы роста. В свою очередь, такие темпы — причина столь же пропорционально высокого роста загрузки (и даже перегрузки) магистральных каналов, собирающих трафик отдельных сетей (локальных, местных и т. п.) Это снижает общую эффективность функционирования сети, заставляя перманентно (постоянно) совершенствовать методы маршрутизации, искать новые подходы к способам, гарантирующим качество сетевого сервиса [54, 55, 56].

Активное использование криптографии в Интернет это, во-первых, — дополнительный и достаточно высокий объем трафика, а, во-вторых, — усложнение протокольной базы. Рассматривая перспективы применения криптографических средств на сетях Интернет необходимо оценивать эффективность их использования с учетом этих факторов. Такая оценка — предмет отдельного, важного и непростого исследования.

Идентификация и аутентификация

Идентификация (именование) и аутентификация (проверка подлинности имени) являются базовыми средствами в реализации ПБ объекта, обеспечивающими для других сервисов безопасности работу с поименованными объектами. Идентификаторы пользователя, от имени которого действует субъект (процесс или пользователь), могут быть разбиты на следующие классы:

- что он знает (пароль, криптографический ключ и т. п.);
- чем он владеет (личную карточку);
- что присуще ему по природе (отпечатки пальцев, голос и т. п.)

Аутентификация для таких классов идентификаторов усложняется сверху вниз, но при этом повышается ее надежность. Процессы, данные (например, криптоключи) или источники данных, которые также могут быть подвергнуты аутентификации, обладают только идентификаторами первого из перечисленных классов. Это обстоятельство облегчает использование криптографии при аутентификации однородных (равноранговых) процессов, в том числе с применением низкоуровневых средств (например, механизмов ядра ОС).

Существует несколько способов — схем реализации службы идентификации/аутентификации. К первой относится, например, традиционная для приложений под ОС UNIX децентрализованная схема, предусматривающая аутентификацию каждого приложения. Отсутствие целостности и централизации в этой схеме затрудняет ее администрирование. Не обеспечивается гибкость в применении различных механизмов (способов) аутентификации, так как в этом случае требуется перекомпиляция приложения.

Второй способ основан на библиотеке встроенных интерактивных модулей PAM (Pluggable Authentication Modules), являющейся частью ОС Red Hat Linux, представляет собой целостную централизованную систему, обеспечивающую гибкое использование различных механизмов аутентификации. Здесь следует отметить, что в разработке PAM вместе с другими участниками рабочей группы Linux-PAM активное участие принимали российские специалисты [39]. Библиотека PAM уже на этапе ее разработки использовалась для создании системы обеспечения безопасности управляющего сегмента сети MSUNet МГУ им. М. В. Ломоносова. Узким местом этого способа аутентификации и библиотеки PAM является обязательное требование интерактивности, что не всегда реализуемо на гетерогенной

сетевой среде, и относительно сложный прикладной интерфейс. Перспективы развития служб идентификации и аутентификации, особенно с учетом их применения для распределенных (информационных и вычислительных) систем, связаны с устранением отмеченных недостатков, а также разработкой новых способов, которые позволяли бы:

- учесть потребности перспективных направлений использования (корпоративный интранет, порталы, распределенные вычислительные метакомпьютерные системы типа GRID [57] и т. п.);
- изыскать (разработать и апробировать) эффективные механизмы аутентификации для отмеченных выше идентификаторов второго и, особенно, третьего классов;
- надежно поддерживать не только традиционную конфиденциальность и целостность данных, но и высокую доступность информации на гетерогенной сетевой среде;
- расширить функциональные возможности системы аутентификации с целью ее более эффективной интеграции с другими сервисами безопасности (разграничение доступа, криптография, протоколирование и аудит и др.);
- использовать механизмы, которые легко реализовать средствами ОС.

К числу эффективных действий на этом направлении следует отнести открытый исследовательский проект PNIAM (Pluggable Non Interactive Authentication Modules [40, 58]), который уже в течение ряда лет ведется по инициативе и при активном участии ученых и преподавателей, студентов и аспирантов механико-математического факультета и Института механики МГУ, научных сотрудников Центра научных телекоммуникаций и информационных технологий РАН. На сегодня многие из результатов (математические модели, алгоритмическое и программное обеспечение), полученные в рамках этого проекта, использованы при создании ОС с повышенными требованиями к надежности серверов на их базе, при разработке дистрибутива для высокопроизводительных вычислительных кластерных систем с удаленным доступом к их ресурсам по Интернет, а также при реализации ряда других практически значимых систем.

Разграничение доступа

Средства логического разграничения доступа определяют действия, которые субъекты (пользователи и процессы) могут выполнять над объектами (информационными ресурсами, процессами, устройствами и т. п.). Такие средства позволяют также обеспечить контроль (поддерживают режим протоколирования) за совершением этих действий. В отличие от физического управления доступом, которое осуществляется на операционном уровне, например, персоналом, регламентирующим доступ пользователя в специальные помещения (компьютерные классы), в данном случае имеется в виду доступ, который обеспечивается программно-техническими средствами. Имея в виду специфику организации Интернет, необходимо подчеркнуть, что этот вид разграничения доступа является определяющим на Метасети.

С формальной точки зрения задача сводится к выполнению заранее установленного (хотя не обязательно статичного) для макрообъекта[2] (продукта или системы ИТ) порядка выполнения для каждого из входящих в макрообъект субъектов операций над каждым из потенциальных атомарных объектов (составляющих макрообъект) при соблюдении, может быть, каких-то дополнительных условий (зависящих, например, от времени, места действия, каких-то ограничений используемого сервиса и т. п.)

Разграничение доступа осуществляется различными аппаратно-программными компонентами от ядра ОС, общецелевых программ (например, графика, СУБД) до отдельных систем прикладного программного обеспечения (например, Web-сервер) на основе принятого порядка выполнения и анализа дополнительных условий.

Модели этих отношений (включая порядок выполнения, дополнительные условия или полномочия), сценарии их формирования и изменения (включая уничтожение или появление новых), способы и механизмы организации, хранения, извлечения и анализа данных являются производными от выбранной

[2]Введение понятия (слова) макрообъект в данном случае (и месте изложения) связано с желанием отделить его от понятия объект в традиционном субъектно-объектном подходе при описании систем логического разграничения доступа.

политики безопасности объекта (продукта или системы ИТ). Они составляют суть (предмет) данного вида сервиса безопасности по отношению к защищаемому объекту. В качестве элементарных, отдельных или атомарных объектов могут выступать файлы, устройства (компьютеры или сетевое оборудование), процессы, такие средства взаимодействия процессов, как сегменты разделяемой памяти, очереди сообщений, семафоры и сокеты, отдельные компоненты прикладных систем, например, таблицы, процедуры баз данных (БД) и т. п. Заметим, что некоторые из них могут выступать (например, в разных видах сервисов или отдельных приложениях) как в роли объектов, так и субъектов (будем обозначать объект/субъект). Такое обилие существующих и постоянно меняющихся субъектов, объектов и отношений между ними объективно затрудняет централизованное логическое управление доступом. В свою очередь отсутствие таких централизованных начал в управлении, что характерно для большинства традиционно используемых систем ИБ, приводит к объективной рассогласованности в адекватном (соответствующем политике безопасности составного макрообъекта) распределении прав и полномочий доступа отдельных субъектов к составляющим (атомарным) объектам при использовании, например, разных видов сервиса. Обмен данными между субъектами/объектами под управлением различных видов сервиса (функционального) на пересечении областей доступа[3] к различным объектам — классический источник «брешей» в системе ИБ различных продуктов или систем ИТ.

В качестве главной (магистральной) цели на пути совершенствования логического управления доступом следует рассматривать подходы к объединению и согласованию на основе общей политики безопасности макрообъекта сценариев, моделей и механизмов такого разграничения на уровнях ОС, отдельных инфраструктурных и функциональных сервисов и приложений. Такое объединение потребует пересмотра многих, в том числе концептуальных, взглядов на подходы к созданию новой модели. Это очень важная, многоплановая задача, конструктивных подходов к решению которой пока не предложено.

Продуктивным на этом направлении может оказаться объектно-ориентированный подход, суть которого применительно к рассматриваемому предмету изложена в [59]. Его реализация сложна, она потребует пересмотра многих уже сложившихся подходов и принципов, длительного времени, однако она объективно необходима. Деятельность на этом направлении способна создать предпосылки к разработке централизованной схемы разграничения доступа в рамках общей системы управления сложными (в том числе распределенными) объектами. Необходимо отметить, что в настоящее время многие из концептуальных подходов, принятых и развиваемых в области ИБ, в том числе критериальная база, развиваются вне объектно-ориентированных представлений. Потребуется большая работа по обоснованию целесообразности (необходимости) нового подхода, строгая математическая формализация задачи (первые соображения изложены в [59]), ее эффективное решение, в том числе — программное, хотя бы на прототипах или ограниченных распределенных прикладных системах, прежде чем он получит признание. Однако с точки зрения перспектив, которые уже рассматриваются, возможности использования современных методов математического моделирования и технологии программирования — это очень интересное предложение, которое заслуживает поддержки.

Развитие систем управления доступом к объектам путем совершенствования используемых (уже существующих) моделей (описывающих правила разграничения доступа, в том числе традиционных — дискреционной и мандатной) за счет комбинации преимуществ каждой из них, использования более гибких схем и тонких механизмов — одна из важнейших задач настоящего времени. Такие работы в мире ведутся, в том числе при поддержке государства в рамках открытых исследовательских проектов. В качестве примеров можно привести проект Trusted Linux [42], направленный прежде всего на изоляцию процессов, выполняющих функции Интернет-серверов и уменьшение последствий атак на них. Пакет LIDS (Linux Intrusion Detection System), созданный в рамках одноименного проекта [43], направленного на создание систем обнаружения вторжений для Linux-систем, позволяет предоставить возможности для более тонкого разделения привилегий между субъектами.

В этом направлении следует отметить исследования, проводимые в 1998–2001 г. на механико-математическом факультете, в Центре телекоммуникаций и технологий Интернет МГУ им М. В. Ломоносова, и практические результаты на их основе по внедрению мандатной политики доступа и совершенствованию механизмов ее реализации на Интернет-серверах, политика безопасности которых предъявляет повышенные требования к их защите [46]. Результаты деятельности на этом направлении совместно

[3]Под областью доступа к объекту здесь будем рассматривать совокупность объектов, доступ которых к нему разрешен.

с другими заинтересованными организациями нашли применение при проектировании и инсталляции Интернет-серверов и создании порталов на сетях государственных ведомств.

Одним из перспективных на ближайшие годы способом доступа к корпоративным информационно-вычислительным ресурсам является Интернет-портал. В архитектуре информационных систем с таким сценарием доступа к ресурсам ключевым звеном, регламентирующим доступ пользователей к информации, выступает Web-сервис и центральный (корневой) Web-сервер. Он является, с одной стороны, информационным концентратором, с другой — первым и единственным рубежом, разграничивающим доступ пользователя к ресурсам. С учетом иерархии такого сорта систем взаимодействие с ресурсами на других нижележащих уровнях происходит с помощью процессов и сервисов (может быть, других — отличных от Web), имеющих опосредованное (от имени пользователя или даже процессов, не имеющих к нему отношения) отношение к пользователю. Взаимодействие на этих уровнях подчиняется правилам разграничения доступа, описанным выше и сталкивается с необходимостью решения задач, о которых уже упоминалось.

Принимая во внимание актуальность и высокую практическую значимость задачи, известную архитектуру и технологические решения Интернет-портала, представляется целесообразным использовать результаты в области развития и интеграции моделей и систем управления доступом, в первую очередь, на этом направлении. С учетом практической важности такой проблемы для крупной корпоративной системы с богатым информационно-вычислительным ресурсом, хороший уровень проработки задачи [49], целесообразно ее поэтапное решение на инфраструктуре одной из сетей-прототипов. В качестве таких могли бы выступать сети MSUNet МГУ им М. В. Ломоносова, RASNet Российской академии наук или, например, другие сети науки и образования.

Протоколирование и аудит

Протоколирование и аудит в системах ИБ обеспечивают возможности для реконструкции прошедших событий и их анализа с целью выявления нарушений, выработки мер к недопущению (исключению) деструктивных воздействий на объект защиты. Степень (объем) применения этого вида сервиса определяется политикой безопасности продукта или системы ИТ. С развитием и усложнением объектов защиты функции этого традиционного вида сервиса значительно расширились. В настоящее время протоколирование и аудит являются базовыми сервисами для формирования так называемых подсистем активного аудита [28, 60]. В условиях отсутствия гарантированно защищенных ОС, невозможности практического пресечения организации скрытых каналов передачи данных, особенно для распределенных систем в Интернет, а также целого ряда других «объективных уязвимостей» в традиционном комплексе средств защиты, подсистемы активного аудита способны существенно повысить уровень безопасности продуктов и систем ИТ. Оперативно анализируя разноплановые результаты протоколов о состоянии подлежащего защите объекта, такая подсистема призвана оперативно обнаружить попытку (потенциальную угрозу) деструктивного воздействия и выработать меры по его предотвращению.

Перспективы создания эффективных подсистем активного аудита в значительной степени связаны с соединением в их составе всех достижений предшествующих продуктов, включая:

- разработки, ориентированные на монокомпьютерные комплексы, где главную роль играют системные сенсоры, средства их анализа и выработки мер для адекватной реакции;
- средства, ориентированные на распределенные структуры, сетевые сенсоры (на основе как пассивных, так и активных методов измерения), механизмы и модели анализа результатов, выработки оперативных мер противодействия деструктивным действиям извне.

Архитектура подобных комплексных систем активного аудита должна быть многоуровневой, где, например, результаты анализа на уровне отдельного компьютера или вычислительного узла, должны дополняться сведениями о состоянии сетевых межкомпьютерных взаимодействий, сетевых сервисов и т. п.

Исследования и разработка подходов к совершенствованию компонент мониторинга состояния подконтрольной системы, механизмов и моделей анализа информации на каждом из ее уровней является очень важным направлением [60].

Описание и программная реализация такой существенно распределенной подсистемы на гетерогенной среде, выполняющей сбор большого объема разноплановых данных представляет собой самостоятельную задачу, соизмеримую по сложности с описанием и программным обеспечением системы в целом. К числу основных задач на этом пути следует отнести:

- описание архитектуры подсистемы, эффективно сочетающей традиционные механизмы протоколирования с нетрадиционными способами организации, оперативного поиска и манипулирования полученными данными;
- исследования и выбор технических средств, алгоритмических решений, способных эффективно реализовать обработку больших объемов данных мониторинга.

К разряду перспективных, с точки зрения повышения эффективности подсистем активного аудита, относится задача, связанная с реализацией механизмов и моделей анализа данных о сетевом трафике, получаемых методами активного мониторинга. На этом направлении необходимо:

- исследовать закономерности в поведении сетевого трафика при «нормальном» режиме функционирования системы, сформулировать математические методы и модели, адекватно описывающие систему в таком состоянии;
- выбрать эффективные способы выявления «отклонений» системы от «нормы», их причины и своевременного реагирования на эти отклонения.

6 Заключение

Отдельные задачи, подходы к их решению, механизмы, модели и инструментальные средства, представленные на обсуждение не претендуют на полноту охвата всех вопросов, связанных с обеспечением информационной безопасности в Интернет. Они формируют лишь общее представление о положении дел и перспективах развития этого направления. С одной стороны, беспрецедентно высокие темпы развития Метасети, расширение сферы ее применения предъявляют новые требования к ее функциональности, управляемости и, в первую очередь — защищенности. С другой стороны, приходится учитывать то обстоятельство, что технологическая база сетей пакетной коммутации на основе стека протоколов TCP/IP изначально не приспособлена для решения задач обеспечения информационной безопасности. На этом фоне многие подходы к реализации защиты информационно-вычислительных ресурсов и сетевой инфраструктуры, лежащие в их основе математические модели и алгоритмические решения требуют серьезного анализа и коренного пересмотра. От того, насколько быстро и эффективно такой пересмотр будет осуществляться, во многом зависит будущее Интернет, как основы мирового информационного пространства. Степень участия, место и роль в таких исследованиях российских специалистов во многом будут определять мощь, независимость и безопасность национального сегмента Интернет. Таких результатов, имеющих мировое звучание и признание, пока мало. Однако исходные положительные предпосылки к успеху на этом пути заключаются в традиционной силе и авторитете российских научных школ в области математики и информатики, результаты последних лет в развитии отечественной телекоммуникационной инфраструктуры и высокопроизводительных вычислений.

Литература

[1] Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model. — ISO/IEC 15408 — 1.1999.

[2] Information technology — Security techniques — Evaluation criteria for IT security — Part 2: Security functional requirements.- ISO/IEC 15408 — 2.1999.

[3] Information technology — Security techniques — Evaluation criteria for IT security — Part 3: Security assurance requirements.- ISO/IEC 15408 — 3.1999.

[4] Проект Госстандарта РФ ГОСТ Р ИСО/МЭК 15408 «Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Части 1, 2, 3. М.: Изд-во Госстандарта России, 2002 г.

[5] Васенин В. А., Галатенко А. В. Компьютерный терроризм и проблемы информационной безопасности в Интернет. В кн.: Высокотехнологичный терроризм. Материалы российско-американского семинара РАН в сотрудничестве с Национальными академиями США. Москва, 4–6 июня 2001 г. М.: 2002, с. 211–225.

[6] Доктрина информационной безопасности Российской Федерации. Утверждена Президентом РФ В. В. Путиным 9 сентября 2000 г.

[7] Гостехкомиссия России. Руководящий документ. Концепция защиты СВТ и АС от несанкционированного доступа к информации. М.: 1992.

[8] Гостехкомиссия России. Руководящий документ. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от НСД к информации. М.: 1992.

[9] Гостехкомиссия России. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. М.: 1992.

[10] Гостехкомиссия России. Руководящий документ. Временное положение по организации разработки, изготовления и эксплуатации программных и технических средств защиты информации от несанкционированного доступа в автоматизированных системах и средствах вычислительной техники. М.: 1992.

[11] Гостехкомиссия России. Руководящий документ. Защита от несанкционированного доступа к информации. Термины и определения. М., 1992.

[12] Грушо А. А., Тимонина Е. Е. Теоретические основы защиты информации. М.: Изд-во агентства «Яхтсмен», 1996, 192 с.

[13] Biba K. J Integrity Consideration for Security Computer System. The MITRE Corp., Report MTR N3153 Revision 1, Electronic System Division, U.S. Air Force Systems Command, Technical Report ESD TR 76 372, Belford, Massachusetts, April 1977.

[14] Грушо А. А., Тимонина Е. Е. Языки в скрытых каналах. Труды международной конференции «Информационные технологии в науке, образовании, телекоммуникациях, бизнесе». Украина, Крым. Ялта — Гурзуф, 19–29 мая 2003 г.

[15] Гайдамакин Н. А. Разграничение доступа к информации в компьютерных системах. Екатеринбург: Изд-во Урал. ун-та, 2003, 328 с.

[16] Бетелин В. Б., Галатенко В. А. Информационная безопасности в России: опыт составления, карты.Jet info, 1998, 1.

[17] Галатенко В. А. Информационная безопасность: практический подход. М.: Наука, 1998, 301 с.

[18] Галатенко В. А. Основы информационной безопасности. Под ред. чл.-корр. РАН В. Б. Бетелина. М.: ИНТУИР.RU, 2003, 280 с.

[19] Зегжда П. Д., Ивашко А. М. Основы безопасности информационных систем. М.: Горячая линия — Телеком, 2000, 452 с.

[20] Щербаков А. Ю. Введение в теорию и практику компьютерной безопасности. М.: Изд. Молгачева С. В., 2001, 352 с.

[21] Bell D. E., LaPadula L. J. Security Computer Systems: Unified Exposition and MULTICS Interpretation, Revision 1, US Air Force ESD-TR-306, MITRE Corporation MTR-2997, Bedford MA, March 1976.

[22] McLean J. Reasoning About Security Models, Proceedings, IEEE Symposium in Privacy and Security, Oakland, CA, April 27–29, 1987, IEEE Computer Society Press, 1987, p. 123–131.

[23] Denning D. et al. Views for multilevel database security, IEEE Trunsaction on Software Engineering, v. SE-13, N 2, 1987, p. 129–140.

[24] Denning D. et al. Multilevel Relational Data Model, Proceedings, IEEE Symposium on Privacy and Security, Oakland, CA, April 27–29, 1987, IEEE Computer Society Press, 1987, p. 220–242.

[25] Goguen J. A., Meseguer J. Security Policies and Security Models. 1982 Symposium on Security and Privacy, p. 11–20, IEEE, April 1982.

[26] Goguen J. A., Meseguer J. Unwinding and Inference Control. 1984, Symposium on Security and Privacy, p. 75–85, IEEE, May 1984.

[27] Галатенко А. В. Об автоматной модели защищенных компьютерных систем. Интеллектуальные системы, т. 4, вып. 3–4, М.: 1999, с. 263–270.

[28] Галатенко А. В. О применении методов теории вероятностей для решения задач информационной безопасности. Вопросы кибернетики. Информационная безопасность. Операционные системы реального времени. Базы данных. М.: НИИСИ РАН, 1999, с. 14–45.

[29] Trusted Computer System Evaluation Criteria, US DOD 5200. 28-STD, December 1985.

[30] National Computer Security Center. A Guide to Understanding Audit in Trusted Systems // NCSC-TG-001, 1987.

[31] National Computer Security Center. A Guide to Understanding Audit in Trusted Systems // NCSC-TG-003, 1987.

[32] National Computer Security Center. Trusted Network Interpretation // NCSC-TG-003, 1987.

[33] Security Architecture for Open Systems Interconnection for CCITT Applications/ Recommendation X.800 // CCITT. Geneva, 1991.

[34] Information Technology Security Evaluation Criteria (ITSEC). Harmonised Criteria of France — Germany — the Netherlands — the United Kindom // Department of trade and Industry. L.: 1991.

[35] Canadian Trusted Computer Product Evaluation Criteria, Version 3.0, Canadian System Security Centre, Communications Security Establishment, Government of Canada, January 1993.

[36] Information Technology Security Evaluation Criteria, Version 1.2, Office for Official Publications of the European Communities, June 1991.

[37] Васенин В. А., Галатенко А. В. О проблемах информационной безопасности в сети Интернет. Глобальная информатизация и безопасность России. Материалы круглого стола «Глобальная информатизация и социально-гуманитарные проблемы человека, культуры и общества». МГУ, октябрь 2000 г.. М.: Изд-во МГУ, 2001, с. 199–214.

[38] Трубачев А. П. и др. Оценка безопасности информационных технологий. Под общ. ред. Галатенко А. В. М.: СИП РИА, 2001.

[39] Савочкин А. В. Комплексный подход к обеспечению безопасности компьютерных систем и сетей, подключенных к Internet. Материалы диссертации на соискание ученой степени кандидата физ.-мат. наук, М.: МГУ, 1999.

[40] Васенин В. А., Галатенко А. В., Савочкин А. В. К построению систем информационной безопасности для научно-образовательных сетей. Труды Всероссийской научной конференции «Научный сервис в сети Интернет», г. Новороссийск, 24–29 сентября 2001 г., Изд-во Моск. ун-та, М.: 2001, с. 178–182.

[41] Васенин В. А, Савкин В. Б. К созданию защищенных систем в Интернет. ICSNET 2001, международный симпозиум по проблемам модельных систем, М.: Изд-во ИЯИ РАН, 2001, с. 245–255.

[42] Dalton C., Tse Huong Choo Trusted Linux: An Operating System Approach to Securing E-Service. Communications of the ACM, 2001, v. 44, issue 2.

[43] Linux IDS Project, `http://www.lids.org`.

[44] Колядов А. Linux 2.4 — шаг к безопасности. Открытые системы, 2001, N 1, с. 22–26.

[45] Галатенко А. В., Лаврентьев А. Ю., Наумов А. А. Создание дистрибутива программного обеспечения для кластеров. Новые информационные технологии в университетском образовании. Тезисы Международной научно-методической конференции, 20–22 марта 2002 г., Кемерово, 2002, с. 192–194.

[46] Галатенко А. В. Реализация многоуровневой политики безопасности в ОС Linux. Информационная безопасность. Инструментальные средства программирования. Микропроцессорные архитектуры. М.: НИИСИ РАН, 2003, с. 107–125.

[47] Безруков В. Л., Годунов А. Н., Назаров П. Е., Солдатов В. А., Хоменков И. И. Введение в ОС 2000. Вопросы кибернетики. Информационная безопасность. Операционные системы реального времени. Базы данных. М.: НИИСИ РАН, 1999, с. 3–13.

[48] POSIX: Information Technology — Portable Operating System Interface (POSIX) —Part 1: Applications Program Interface (API). ISO/IEC 9945-1.

[49] Васенин В. А., Афонин С. А., Коршунов А. А. К созданию концепции интегрированной системы распределенных информационных ресурсов Московского государственного университета им. М. В. Ломоносова, М.: Изд-во Моск. ун-та, 2001, 112 с.

[50] Васенин В. А., Корнеев В. В., Ландина М. Ю., Роганов В. А. Система функционального активного мониторинга FLAME. Программирование, 2003, N 3, с. 161–173.

[51] Введение в криптографию. Под ред. В. В. Ященко. М.: МЦНМО — ЧеРо, 2000.

[52] Московский университет и развитие криптографии в России. Материалы конференции в МГУ 17–18 октября 2002 г. М.: МЦНМО, 2003, 287 с.

[53] Козлов В. А. Открытые информационные системы. М.: Финансы и статистика, 1999, 224 с.

[54] Васенин В. А. Научно-образовательные сети, новые информационные технологии и приложения. Сб. «Логика и приложения». Тезисы международной конференции, посвященной 60-летию со дня рождения акад. Ю. Л. Ершова. Новосибирск, 4–6 мая 2000 г., с. 24–28.

[55] Виняр Д. Анализ современных методов маршрутизации. Jet Jnfo, 1998, N 2–3 (57–58).

[56] Васенин В. А. Высокопроизводительные научно-образовательные сети России. Настоящее и будущее. М.: Изд-во Моск. ун-та, 1999, 32 с.

[57] Васенин В. А., Жижченко А. Б. Математические модели, алгоритмы и программное обеспечение информационных систем нового поколения. Высокопроизводительные вычисления и технологии. Тезисы Всероссийской конференции. Москва — Ижевск: Институт компьютерных исследований, 2003, с. 28–34.

[58] Галатенко А. В. Реализация сервисов безопасности на основе встраиваемых модулей. Информационная безопасность. Инструментальные средства программирования. Микропроцессорные архитектуры. М.: НИИСИ РАН, 2003, с. 91–106.

[59] Галатенко А. В., Галатенко В. А. О постановке задачи разграничения доступа в распределенной объектной среде. Сб. «Вопросы кибернетики». Информационная безопасность. Операционных системы реального времени. Базы данных. Под ред. В. Б. Бетелина, М.: Изд-во ВИНИТИ, 1999, с. 3–13.

[60] Галатенко А. В. Активный аудит. Jet Info, 1999,N 8.

Информационная безопасность электронного бизнеса

В. Д. Аносов, А. С. Кузьмин

В настоящее время в Российской Федерации в рамках ряда федеральных целевых программ проводится комплекс работ по внедрению современных информационных технологий в деятельность органов государственной власти, органов местного самоуправления и хозяйствующих субъектов. При этом предусматривается перевод в электронную цифровую форму большей части документооборота, осуществляемого между хозяйствующими субъектами, органами государственной власти и органами местного самоуправления. Проблема обеспечения безопасности экономически значимой информации осложняется ввиду использования сети Интернет в качестве основной транспортной среды электронного документооборота. В соответствии с «Доктриной информационной безопасности Российской Федерации», воздействию угроз информационной безопасности Российской Федерации в сфере экономики наиболее подвержены:

- система государственной статистики;
- кредитно-финансовая система;
- информационные и учетные автоматизированные системы подразделений федеральных органов исполнительной власти, обеспечивающих деятельность общества и государства в сфере экономики;
- системы бухгалтерского учета предприятий, учреждений и организаций независимо от формы собственности;
- системы сбора, обработки, хранения и передачи финансовой, биржевой, налоговой, таможенной информации и информации о внешнеэкономической деятельности государства, а также предприятий, учреждений и организаций независимо от формы собственности.

Для достижения необходимой степени защищенности информации, циркулирующей в сфере экономики, необходимо решить как общие для обеспечения информационной безопасности современных информационно-телекоммуникационных систем вопросы: целостность, конфиденциальность, доступность,так и вопросы, специфичные дляинформационно-телекоммуникационных систем экономической сферы: обеспечение юридической значимости электронных документов, обеспечение неотслеживаемости электронных платежей, подтверждение подлинности и обязательств электронных транзакций. В сфере экономики имеется ряд особенностей, которые оказывают существенное влияние на выбор механизмов обеспечения информационной безопасности. Среди них можно отметить следующие:

- специфичная модель угроз и нарушителя;
- неоднородность коммерческих организаций;
- возможность страхования информационных рисков;
- необходимость определения ценности информации;
- целесообразность динамичности системы защиты и ее мониторинга;
- применение открыто созданных средств защиты и т. д.

Для повышения безопасности экономической информации и информационных ресурсов российской экономики важное значение имеет использование отечественных криптографических примитивов в

прикладных протоколах. При их разработке имеет значение как обеспечение функций безопасности, реализуемых в прикладном протоколе, так и учет особенностей бизнес-процесса, для которого рекомендуется использовать данный прикладной протокол.

Можно выделить несколько секторов экономической деятельности, реализуемых в электронной форме:

1. Выполнение заказов для государственных учреждений — B2G (Business to Government);
2. Обеспечение взаимодействия в электронной форме при участии в нем экономически значимых субъектов. При этом наибольший интерес представляет такая его разновидность, как B2B (Business to Business) — «бизнес для бизнеса», межкорпоративный сегмент электронной коммерции;
3. Бизнес-модели, ориентированные на конечного пользователя, — B2C (Business to Consumer).

На основе анализа криптографических протоколов, используемых в системах электронной коммерции, электронной торговли, кредитно-финансовой сферы, электронного документооборота даются рекомендации по их использованию и пополнению перечня отечественных стандартизированных криптографических примитивов.

Часть III
Секционные доклады

Спектр в нехемминговской метрике 4-значного кода, порожденного фукциями $\mathrm{Tr}_4(ax^5+bx)$[1]

В. М. Сидельников, И. Б. Гашков

1 Нехемминговская метрика $\widehat{\lambda}$ на $\mathbb{F}_4$

Мы рассматриваем конечное поле $\mathbb{F}_4 = \{0, 1, \omega, \omega'\}$, где $\{\omega, \omega'\}$ — корни неприводимого многочлена $x^2 + x + 1$. Аддитивная группа $\mathfrak{G}$, $|\mathfrak{G}| = 4$, этого поля изоморфна элементарной абелевой 2-группе $\mathbb{F}_2^2$. Каждый элемент $\mathbb{F}_4$ мы будем трактовать как двоичный вектор размерности 2, координаты которого являются коэффициентами в его представлении в нормальном базисе $\{\omega, \omega'\}$ поля $\mathbb{F}_4$ над полем $\mathbb{F}_2$. Это двумерное представление элемента $a \in \mathbb{F}_4$ группы $\mathfrak{G}$ будем обозначать через $\psi(a)$. Например, для элемента $1 \in \mathbb{F}_4$ $\psi(1) = (1, 1)$, ибо $1 = \omega + \omega'$.

На группе $\mathfrak{G}$ с групповой операцией $+$ мы определим нехемминговскую метрику $\widehat{\lambda}$ (известную также под названием метрика Ли [1]) с помощью соотношения

$$\widehat{\lambda}(a, b) = d(\psi(a), \psi(b)), \tag{1}$$

где d — растояние Хемминга на $\mathbb{F}_2^2$. Функция $\widehat{\lambda}(a, b)$ принимает три значения: $0, 1, 2$. Как легко проверить, функция $\widehat{\lambda}$ действительно является метрикой на $\mathfrak{G}$.

На группе $\mathfrak{G}^m$ мы определим нехемминговскую метрику $\widehat{\lambda}$, положив $\widehat{\lambda}(\boldsymbol{a}, \boldsymbol{b}) = \sum_{j=1}^{m} \widehat{\lambda}(a_j, b_j)$, где $\boldsymbol{a} = (a_1, \ldots, a_m), \boldsymbol{b} \in \mathfrak{G}^m$. Отображение ψ элементов $\mathfrak{G}^m$ в элементы $\mathbb{F}_2^{2m}$ определим как $\psi(\boldsymbol{a}) = (\psi(a_1), \ldots, \psi(a_n))$. Код $\mathfrak{K} \subseteq \mathbb{F}_4^m$ длины m с кодовым расстоянием $\widehat{\lambda}(\mathfrak{K})$ в метрике $\widehat{\lambda}$ отображением ψ переводится в двоичный код в $\mathbb{F}_2^{2m}$ длины $2m$ с тем же распределением взаимных расстояний, в частности, с тем же кодовым расстоянием.

Очевидно, $\widehat{\lambda}$ — транзитивно-инвариантная метрика, т. е. $\widehat{\lambda}(\boldsymbol{a}, \boldsymbol{b}) = \widehat{\lambda}(\boldsymbol{a}+\boldsymbol{c}, \boldsymbol{b}+\boldsymbol{c})$ при любом $\boldsymbol{c} \in \mathfrak{G}^m$. Поэтому метрика $\widehat{\lambda}$ определяется функцией $\mathrm{wt}(\boldsymbol{a}) = \widehat{\lambda}(\boldsymbol{a}, \boldsymbol{0})$: $\widehat{\lambda}(\boldsymbol{a}, \boldsymbol{b}) = \mathrm{wt}(\boldsymbol{a}-\boldsymbol{b})$. Функцию $\mathrm{wt}(\boldsymbol{a})$ будем называть нехемминговским весом вектора $\boldsymbol{a}$.

Функция $\widehat{\lambda}$ действительно является метрикой на $\mathfrak{G}^m$, ибо она превращается в метрику Хемминга на двоичном пространстве $\mathbb{F}_2^{2m}$, которое является образом $\mathfrak{G}^m$ при его отображении ψ.

Пусть $\mathbb{F}_q$, $q = 2^n$, — конечное поле. В том случае, когда $u \mid n$, через $\mathrm{Tr}_{2^u}^{(q)}$ мы обозначаем функцию след, определенную соотношением $\mathrm{Tr}_{2^u}^{(q)}(x) = x + x^{2^u} + x^{2^{2u}} + \ldots + x^{2^{n-u}}$ ($\frac{n}{u}$ слагаемых). Будем рассматривать только случаи $u = 1, 2, 4$. Очевидно, $\mathrm{Tr}_2^{(q)} = \mathrm{Tr}_4^{(q)}(x) + \left(\mathrm{Tr}_4^{(q)}\right)^2(x)$, $\mathrm{Tr}_4^{(q)} = \mathrm{Tr}_{16}^{(q)}(x) + \left(\mathrm{Tr}_{16}^{(q)}\right)^4(x)$. Верхний индекс у $\mathrm{Tr}_2^{(q)}$ и $\mathrm{Tr}_4^{(q)}$ будем иногда опускать.

Следующая лемма очевидна

Лемма 1. *Пусть $a \in \mathbb{F}_4$. Имеет место равенство*

$$\mathrm{wt}(a) = \frac{1}{2}\left(2 - (-1)^{\mathrm{Tr}_2^{(4)}(a\omega)} - (-1)^{\mathrm{Tr}_2^{(4)}(a\omega')}\right). \tag{2}$$

Следствие 1. *Пусть $\boldsymbol{a} \in \mathbb{F}_4^N$, тогда*

$$\mathrm{wt}(\boldsymbol{a}) = N - \frac{1}{2}\sum_{i=1}^{N}\left((-1)^{\mathrm{Tr}_2^{(4)}(a_i\omega)} + (-1)^{\mathrm{Tr}_2^{(4)}(a_i\omega')}\right). \tag{3}$$

[1]Работа поддержаны Российским Фондом Фундаментальных Исследований (грант № 02-01-00687) и фондом INTAS (grant № 00-738).

В последующем мы вычислим в метрике $\widehat{\lambda}$ спектр 4-значного кода $\mathfrak{K}_4$ длины q, порожденного всеми векторами вида $\boldsymbol{\alpha}_f = (\mathrm{Tr}_4(f(\alpha_1)), \ldots, \mathrm{Tr}_4(f(\alpha_q))$, $f(x) = ax^3 + bx$, $a, b \in \mathbb{F}_q$, где $\{\alpha_1, \ldots, \alpha_q\} = \mathbb{F}_q$. Это позволит нам вычислить и спектр двоичного кода $\psi(\mathfrak{K}_4)$ длины $2q$.

2 Квадратичные формы над $\mathbb{F}_4$

Рассмотрим квадратичную форму

$$F(\boldsymbol{x}) = F(x_1, \ldots, x_n) = \sum_{1 \leqslant i \leqslant j \leqslant n} a_{i,j} x_i x_j \tag{4}$$

с коэффициентами из поля $\mathbb{F}_4$.

Лемма 2. *Квадратичную форму $F(x_1, \ldots, x_n)$ с помощью невырожденного линейного преобразования переменных $\boldsymbol{x} = \boldsymbol{x}'A$ можно привести к виду*

$$F(x_1, \ldots, x_n) = \sum_{i=1}^{t} x'_{2i-1} x'_{2i} + \sum_{i=2t+1}^{n} a_i x_i'^2 = F'(x'_1, \ldots, x'_n) + \sum_{i=2t+1}^{n} a_i x_i'^2, \tag{5}$$

где $t \leqslant \left[\frac{n}{2}\right]$.

Замечание. Авторы затрудняются указать публикацию, в которой рассматривались бы квадратичные формы над $\mathbb{F}_4$ данного вида.

Число $2t$ будем называть рангом формы $F(\boldsymbol{x})$.

Квадратичную функцию $f(x, y) = xy + cx + dy$ с помощью аффинного преобразования $x = x' + d$, $y = y' + c$ можно привести к виду $f(x, y) = x'y' + cd$. Отсюда и из леммы 2 вытекает

Теорема 1. *Квадратичную функцию $f(\boldsymbol{x}) = F(x_1, \ldots, x_n) + l(x_1, \ldots, x_n)$, где $l(x_1, \ldots, x_n) = \langle \boldsymbol{l}, \boldsymbol{x} \rangle$ — линейная функция, с помощью невырожденного аффинного преобразования $\boldsymbol{x} = \boldsymbol{x}'A + \boldsymbol{l}A^T$ переменных можно привести к виду*

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{t} x'_{2i-1} x'_{2i} + \sum_{i=2t+1}^{n} (b_i x_i'^2 + c_i x'_i) + F(\boldsymbol{l}), \tag{6}$$

где $t \leqslant \left[\frac{n}{2}\right]$.

Пусть $x \in \mathbb{F}_4$. Тогда

$$\chi(x) = \frac{1}{4}(1 + (-1)^{\mathrm{Tr}_2(a\omega)}) \left(1 + (-1)^{\mathrm{Tr}_2(a\omega')}\right) = \begin{cases} 1, & \text{если } x = 0 \\ 0, & \text{если } x \neq 0 \end{cases}, \tag{7}$$

где $\mathrm{Tr}_2(x) = x^2 + x$ — след элемента x поля $\mathbb{F}_4$ в поле $\mathbb{F}_2$. Отсюда вытекает

Лемма 3. *Число N_f нулевых значений, принимаемых функцией $f(\boldsymbol{x})$, когда $\boldsymbol{x} = (x_1, \ldots, x_n)$ пробегает все элементы пространства $\mathbb{F}_4^n$, равно*

$$\begin{aligned} N_f &= \frac{1}{4} \sum_{\boldsymbol{x} \in \mathbb{F}_4^n} \left(1 + (-1)^{\mathrm{Tr}_2(\omega f(\boldsymbol{x}))}\right) \left(1 + (-1)^{\mathrm{Tr}_2(\omega' f(\boldsymbol{x}))}\right) \\ &= 4^{n-1} + \frac{1}{4} \sum_{\gamma \in \mathbb{F}_4^*} \sum_{\boldsymbol{x} \in \mathbb{F}_4^n} (-1)^{\mathrm{Tr}_2(\gamma f(\boldsymbol{x}))}. \end{aligned} \tag{8}$$

Доказательство леммы 3 следует из (7).

Следствие 2. *Если функция $f(\boldsymbol{x})$ в (6) обладает тем свойством, что $\mathrm{Tr}_2(\sum_{i=2t+1}^{n} b_i x_i'^2 + \sum_{i=2t+1}^{n} c_i x'_i)$ не равна тождественно 0, то $N_F = 4^{n-1}$.*

Заметим, что функция $\mathrm{Tr}_2(\sum_{i=2t+1}^{n} b_i x_i'^2 + \sum_{i=2t+1}^{n} c_i x_i')$ тождественно равна 0 тогда и только тогда, когда $\sum_{i=2t+1}^{n} b_i x_i'^2 + \sum_{i=2t+1}^{n} c_i x_i' = \sum_{i=2t+1}^{n} \varepsilon_i (x_i'^2 + x_i')$, $\varepsilon_i \in \{0,1\}$.

Следствие 3. *Если в* (6) $\sum_{i=2t+1}^{n} b_i x_i'^2 + \sum_{i=2t+1}^{n} c_i x_i' = \sum_{i=2t+1}^{n} \varepsilon_i (x_i'^2 + x_i')$, $\varepsilon_i \in \{0,1\}$, *то*

$$N_f = 4^{n-1} + 4^{n-2t} \begin{cases} 3 \cdot 4^{t-1}, & \text{если } F(\boldsymbol{l}) = 0 \\ -4^{t-1}, & \text{если } F(\boldsymbol{l}) \neq 0 \end{cases}. \tag{9}$$

Пусть $\mathbb{F}_q$, $q = 4^n$, — конечное поле и $\mathrm{Tr}_4(x) = x + x^4 + \ldots + x^{4^{n-1}}$ — след элемента $x \in \mathbb{F}_q$ в поле $\mathbb{F}_4$. Пусть $\Omega = \{\omega_1, \ldots, \omega_n\}$ — базис поля $\mathbb{F}_q$ над полем $\mathbb{F}_4$. Запишем элемент $x \in \mathbb{F}_q$ в виде $x = x_1\omega_1 + \ldots + x_n\omega_n$, $x_j \in \mathbb{F}_4$. В этом случае функцию $\mathrm{Tr}_4(ax^5)$, $a \in \mathbb{F}_q$, можно записать в виде $\mathrm{Tr}_4(x^5) = \sum_{i=1}^{n} \mathrm{Tr}_4(a^5(x_1\omega_1 + \ldots + x_n\omega_n)(x_1\omega_1 + \ldots + x_n\omega_n)^4) = \sum_{i=1}^{n} \mathrm{Tr}_4(a^5(x_1\omega_1 + \ldots + x_n\omega_n)(x_1\omega_1^4 + \ldots + x_n\omega_n^4)) = \sum_{1 \leqslant i \leqslant j \leqslant n} a_{i,j} x_i x_j$, где $a_{i,j} = \mathrm{Tr}_4(a^5 \omega_i \omega_j^4)$.

Таким образом, функция $F(x_1, \ldots, x_n) = \mathrm{Tr}_4(ax^5)$ является квадратичной формой вида (4), а функция $f(\boldsymbol{x}) = \mathrm{Tr}_4(ax^5 + bx)$ — квадратичной функцией вида $f(\boldsymbol{x}) = F(\boldsymbol{x}) + \langle \boldsymbol{l}, \boldsymbol{x} \rangle$.

3 Ранг функции $\mathrm{Tr}_4(ax^5 + bx)$

Лемма 4. *Пусть* $a \neq 0, a, b \in F_q$ *и* $T_{a,b} = \sum_{x \in \mathbb{F}_q} (-1)^{\mathrm{Tr}_2(ax^5 + bx)}$, *где* $\mathrm{Tr}_2 = \mathrm{Tr}_2^{(q)}$. *Тогда при нечетном* n

$$|T_{a,b}| = \begin{cases} 4^{\frac{n+1}{2}}, & \text{если } \mathrm{Tr}_4^{(q)}\left(ba^{-\frac{1}{5}}\right) = 1 \\ 0, & \text{если } \mathrm{Tr}_4^{(q)}\left(ba^{-\frac{1}{5}}\right) \neq 1 \end{cases}, \tag{10}$$

и при четном n

$$|T_{a,b}| = \begin{cases} 4^{\frac{n+2}{2}}, & \text{если } a^{\frac{4^n-1}{5}} = 1 \text{ и } \mathrm{Tr}_4^{(q)}\left(ba^{-\frac{1}{5}}\right) = 0 \\ 0, & \text{если } a^{\frac{4^n-1}{5}} = 1 \text{ и } \mathrm{Tr}_4^{(q)}\left(ba^{-\frac{1}{5}}\right) \neq 0 \\ 4^{\frac{n}{2}}, & \text{если } a^{\frac{4^n-1}{5}} \neq 1 \end{cases}. \tag{11}$$

Доказательство. Имеем

$$\begin{aligned} T_{a,b}^2 &= \sum_{x,y \in \mathbb{F}_q} (-1)^{\mathrm{Tr}_2(ax^5 + ay^5 + b(x+y))} = \sum_{x,z \in \mathbb{F}_q} (-1)^{\mathrm{Tr}_2(a(x^5 + (x+z)^5) + bz)} \\ &= \sum_{x,z \in \mathbb{F}_q} (-1)^{\mathrm{Tr}_2(a(x^4 z + x z^4 + z^5) + bz)} = \sum_{x,z \in \mathbb{F}_q} (-1)^{\mathrm{Tr}_2(a(x^4(z + a^3 z^{16}) + z^5) + bz)} \\ &= 4^n \sum_{z + a^3 z^{16} = 0} (-1)^{\mathrm{Tr}_2(az^5 + bz)} \end{aligned} \tag{12}$$

Если z_0 — ненулевой корень уравнения $z + a^3 z^{16} = 0$ в поле $\mathbb{F}_q$, то остальными корнями в поле $\mathbb{F}_q$ этого уравнения являются элементы

$$\gamma z_0, \quad \gamma \in \begin{cases} \mathbb{F}_4, & \text{если } n \text{ — нечетное число} \\ \mathbb{F}_{16}, & \text{если } n \text{ — четное число} \end{cases}.$$

Ввиду того, что числа 5 и $4^n - 1$ взаимно просты, уравнение $z + a^3 z^{16} = 0$ при нечетном n всегда имеет ненулевое решение $z_0 = a^{-\frac{1}{5}}$ в поле $\mathbb{F}_q$.

Если n — четное число, то уравнение $z + a^3 z^{16} = 0$ имеет ненулевое решение $z_0 = a^{-\frac{1}{5}}$ тогда и только тогда, когда $a^{\frac{4^n-1}{5}} = 1$.

Заметим, что

$$\mathrm{Tr}_2(x) = \begin{cases} \mathrm{Tr}_2^{(4)}(\mathrm{Tr}_4^{(q)}(x)), & \text{если } n \text{ — нечетное число} \\ \mathrm{Tr}_2^{16}(\mathrm{Tr}_{16}^{(q)}(x)), & \text{если } n \text{ — четное число} \end{cases}.$$

Поэтому при нечетном n из (12) вытекает

$$T_{a,b}^2 = 4^n \sum_{\gamma\in\mathbb{F}_4} (-1)^{\mathrm{Tr}_2^{(4)}\left(\gamma^2\,\mathrm{Tr}_4^{(q)}(1)+\gamma\,\mathrm{Tr}_4^{(q)}\left(ba^{\frac{1}{5}}\right)\right)} = \begin{cases} 4, & \text{если } \mathrm{Tr}_4^{(q)}\left(ba^{-\frac{1}{5}}\right) = 1 \\ 0, & \text{если } \mathrm{Tr}_4^{(q)}\left(ba^{-\frac{1}{5}}\right) \neq 1 \end{cases}, \tag{13}$$

ибо в рассматриваемом случае $\mathrm{Tr}_4^{(q)}(1) = 1$.

При четном n из (12) вытекает

$$T_a^2 = \begin{cases} 4^n \sum_{\gamma\in\mathbb{F}_{16}} (-1)^{\mathrm{Tr}_2\left(\gamma^5\,\mathrm{Tr}_{16}^{(q)}(1)+\gamma\,\mathrm{Tr}_{16}^{(q)}\left(ba^{-\frac{1}{5}}\right)\right)}, & \text{если } a^{\frac{4^n-1}{5}} = 1 \\ 4^n, & \text{если } a^{\frac{4^n-1}{5}} \neq 1 \end{cases}. \tag{14}$$

Заметим, что $\mathrm{Tr}_{16}^{(q)}(1) = 0$ или $\mathrm{Tr}_{16}^{(q)}(1) = 1$. Если $\gamma \in \mathbb{F}_{16}$, то $\gamma^5 \in \mathbb{F}_4$. Следовательно, $\mathrm{Tr}_2^{(16)}(\gamma^5\,\mathrm{Tr}_{16}^{(q)}(1)) = 0$ при любом $\gamma \in \mathbb{F}_{16}$. Поэтому

$$\sum_{\gamma\in\mathbb{F}_{16}} (-1)^{\mathrm{Tr}_2\left(\gamma^5\,\mathrm{Tr}_{16}^{(q)}(1)+\gamma\,\mathrm{Tr}_{16}^{(q)}\left(ba^{-\frac{1}{5}}\right)\right)} = \begin{cases} 16, & \text{если } \mathrm{Tr}_{16}^{(q)}\left(ba^{-\frac{1}{5}}\right) = 0 \\ 0, & \text{если } \mathrm{Tr}_{16}^{(q)}\left(ba^{-\frac{1}{5}}\right) \neq 0 \end{cases}. \tag{15}$$

Из полученных соотношений вытекает утверждение леммы. □

4 Линейный код

Пусть $\mathbb{F}_q = \{\alpha_1, \ldots, \alpha_q\}$, $q = 4^n$, где n, $n \geqslant 3$, — нечетное число, и $f = f(x) \in \mathbb{F}_4[x]$, — многочлен над $\mathbb{F}_4$. Положим

$$\boldsymbol{x}_f = \left(\mathrm{Tr}_4^{(q)}(f(\alpha_1)), \ldots, \mathrm{Tr}_4^{(q)}(f(\alpha_q))\right) \tag{16}$$

Рассмотрим линейный над $\mathbb{F}_4$ код $\mathfrak{K}^{(q)} = \left\{\boldsymbol{x}_f \;\middle|\; f \in \widetilde{\mathbb{F}}_4[x]\right\}$ над $\mathbb{F}_4$ длины q, где множество $\widetilde{\mathbb{F}}_4[x]$ образовано многочленами вида $f(x) = ax^5 + bx + c$, $a, b \in \mathbb{F}_q$, $c \in \mathbb{F}_4$. Код $\mathfrak{K}^{(q)}$, очевидно, имеет размерность $k = 2n + 1$.

Теорема 2. *Линейный над $\mathbb{F}_4$ код $\mathfrak{K}^{(q)}$ имеет спектр (в метрике $\widehat{\lambda}$), который приведен в таблице 1. В частности, его кодовое расстояние $\widehat{\lambda}_{\min} = \widehat{\lambda}\left(\mathfrak{K}^{(q)}\right)$ равно*

$$\widehat{\lambda}_{\min} = q - \sqrt{q}. \tag{17}$$

No	вес wt	число элементов с данным весом
1	0	1
2	$2q$	1
3	q	$(q-1)q+2$
4	$q+\sqrt{q}$	$(3q^2+q-4)/2$
5	$q-\sqrt{q}$	$(3q^2+q-4)/2$

Таблица 1:

Доказательство. Пусть $f(x) = ax^5 + bx$. Воспользуемся соотношением (1). Имеем

$$\begin{aligned} \mathrm{wt}(\boldsymbol{x}_f) &= q - \frac{1}{2}\sum_{i=1}^{q}\left((-1)^{\mathrm{Tr}_2^{(4)}\left(\omega\,\mathrm{Tr}_4^{(q)}(f(\alpha_1))\right)} + (-1)^{\mathrm{Tr}_2^{(4)}\left(\omega'\,\mathrm{Tr}_4^{(q)}(f(\alpha_1))\right)}\right) \\ &= q - \frac{1}{2}\sum_{x\in\mathbb{F}_q}\left((-1)^{\mathrm{Tr}_2^{(q)}(\omega(ax^5+bx+c))} + (-1)^{\mathrm{Tr}_2^{(q)}(\omega'(ax^5+bx+c))}\right) \\ &= q - \frac{1}{2}(T_{\omega a,\omega b} + T_{\omega' a,\omega' b}). \end{aligned} \tag{18}$$

Покажем, что при a, b не равных одновременно 0 функция $\tau_{a,b} = |(T_{\omega a,\omega b} + T_{\omega' a,\omega' b})|$, a, b принимает одно из двух значений $0, 2\sqrt{q}$.

Имея в виду равенство (10), достаточно показать, что $|T_{\omega a,\omega b}|$ и $|T_{\omega' a,\omega' b}|$ не могут одновременно принимать значение $2\sqrt{q}$. Для этого мы покажем, что если $|T_{\omega a,\omega b}| = 2\sqrt{q}$, то $|T_{\omega' a,\omega' b}| = 0$.

Действительно, если $\mathrm{Tr}_4^{(q)}\left(\omega b(\omega a)^{-\frac{1}{5}}\right) = 1$, то $\mathrm{Tr}_4^{(q)}\left(\omega' b(\omega' a)^{-\frac{1}{5}}\right) \neq 1$. Последнее вытекает из (10) и из того, что $\omega^{\frac{4}{5}} \neq \omega'^{\frac{4}{5}}$.

Вычисление спектра. Как следует из (10) и того, что при $|T_{\omega a,\omega b}| \neq 0$ $|T_{\omega' a,\omega' b}| = 0$, функция $\tau_{a,b}$ принимает значение 0 тогда и только тогда, когда $\mathrm{Tr}_4^{(q)}\left(ba^{-\frac{1}{5}}\right) = 1$. Число решений a, b, c, $(a, b \in \mathbb{F}_q$, $c \in \mathbb{F}_4)$ последнего уравнения, очевидно равно $(q-1)q$. Таким образом, число векторов $\boldsymbol{x}_{a,b}$, для которых $\mathrm{wt}(\boldsymbol{x}_{a,b}) = q$ равно $(q-1)q+2$, где 2 — число многочленов $f(x) = ax^5 + bx + c$, $a = b = 0$, $c = \omega, \omega'$, которые определяют векторы $\boldsymbol{x}_{a,b}$ веса q.

Легко установить, что $\frac{1}{|\mathfrak{K}^{(q)}|}\sum_{\boldsymbol{x}\in\mathfrak{K}^{(q)}} \mathrm{wt}(\boldsymbol{x}) = q$. Если обозначить через A_+ и A_- число векторов веса $q \pm \sqrt{q}$, через A_0, $A_q = (q-1)q+2$, A_{2q} число векторов веса 0, q и $2q$, то из последнего соотношения вытекает, $\frac{1}{|\mathfrak{K}^{(q)}|}\sum_{\boldsymbol{x}\in\mathfrak{K}^{(q)}} \mathrm{wt}(\boldsymbol{x}) = \frac{1}{|\mathfrak{K}^{(q)}|}(2q + q((q-1)q+2) + (q+\sqrt{q})A_+ + (q-\sqrt{q})A_-) = q$. Кроме того, $A_0 + A_{2q} + A_q + A_+ + A_- = |\mathfrak{K}^{(q)}|$. Из последних двух уравнений с неизвестными A_+, A_- следует, что $A_+ = (3q^2+q-4)/2$, $A_- = (3q^2+q-4)/2$. □

Если применить к координатам векторов кода $\mathfrak{K}^{(q)}$ отображение ψ, то мы получим двоичный линейный код $\psi\left(\mathfrak{K}^{(q)}\right)$ длины $N = 2q = 2^{2n+1}$ с кодовым расстоянием Хемминга $d = \lambda = N/2 - \sqrt{N/2}$ и числом элементов N^2. Спектр кода $\psi\left(\mathfrak{K}^{(q)}\right)$ приведен в таблице 2.

No	вес wt	число элементов с данным весом
1	0	1
2	N	1
3	$N/2$	$(N-2)N/4+2$
4	$N/2+\sqrt{N/2}$	$(3N^2+2N-16)/8$
5	$N/2-\sqrt{N/2}$	$(3N^2+2N-16)/8$

Таблица 2:

Литература

[1] Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов, исправляющих ошибки. М.: Связь, 1979.

[2] Хьюзмоллер Д., Милнор Дж. Симметрические билинейные формы. М.: Наука, 1986.

[3] Лидл Р., Нидеррайтер Г. Конечные поля. Т 1, 2. М.: Мир, 1988.

Математические модели помехоустойчивости и помехозащищённости квантовых компьютеров

В. А. Винокуров, В. А. Садовничий

Главный элемент идеального квантового компьютера — электрон, находящийся в определённом квантовом состоянии. Поэтому проектирование квантового компьютера и его характеристик опирается на изучение поведения одного электрона во внешнем электростатическом поле. Простейшая математическая модель этой ситуации — краевая задача для одномерного стационарного уравнения Шредингера вида

$$\psi'' + (\lambda - q(x))\psi = 0 \qquad (1)$$

на отрезке $[0,\ell]$ с граничными условиями $\psi(0)=\psi(\ell)=0$. Здесь $\psi(x)$ — пси-функция, нормированная условием $\int_0^\ell |\psi(x)|^2\,dx=1$, собственное значение $\lambda=\frac{2m}{\hbar^2}E$, E — энергия электрона, функция $q(x)=\frac{2me}{\hbar^2}\varphi(x)$, функция $\varphi(x)$ — потенциал внешнего электростатического поля, m — масса электрона, e — заряд электрона, $\hbar=h/2\pi$, h — постоянная Планка.

Если возмущение отсутствует, т. е. $\varphi\equiv 0$, то первое собственное значение, соответствующее основному состоянию электрона, $\lambda_{1,0}=\left(\frac{\pi}{\ell}\right)^2$, что соответствует энергии основного состояния $E_1=\frac{1}{2m}\left(\frac{\hbar\pi}{\ell}\right)^2$.

Рассматривается вопрос: как сильно изменяется энергия основного состояния при наличии внешнего электромагнитного возмущения, задаваемого потенциалом $\varphi(x)$. Достаточно просто устанавливаются следующие границы изменения собственного значения

$$|E_1-E_{1,0}|\leqslant e\|\varphi\|_\infty, \tag{2}$$

где $\|\varphi\|_\infty\equiv \operatorname{vrai}\sup_{x\in[0,\ell]}|q(x)|$. Более сложными рассуждениями устанавливаются оценки сдвига собственного значения через норму потенциала $\|\varphi\|_1\equiv\int_0^\ell|\varphi(x)|\,dx$ в пространстве L_1. Оценка сдвига вверх

$$E_1-E_{1,0}\leqslant 2\frac{e}{\ell}\|\varphi\|_1. \tag{3}$$

Оценки сдвига вниз

$$E_1\geqslant -\frac{me^2}{2\hbar^2}\|\varphi\|_1^2. \tag{4}$$

И оценка сдвига вниз при $\|\varphi\|_1\leqslant\frac{2\hbar^2}{me\ell}$ вида

$$E_1-E_{1,0}\geqslant -\left(\frac{\pi}{2}\right)^2\frac{e}{\ell}\|\varphi\|_1. \tag{5}$$

Полное изложение соответствующих математических вопросов можно найти по адресу `http://vinokur.narod.ru/eval.html`.

Новый подход к безусловной секретности в релятивистской квантовой криптографии

С. Н. Молотков

Предлагается принципиально новый подход к обеспечению секретного распространения ключа по открытым квантовым каналам связи. В отличие от предыдущих схем, где секретность основана на специальных свойствах неортогональных состояний в гильбертовом пространстве, секретность в предлагаемой схеме базируется на пространственно-временной структуре состояний и ограничениях, диктуемых специальной теорий относительности. Учет этих обстоятельств позволяет передавать секретный ключ при помощи практически любых квантовых состояний. Получены ограничения на допустимый поток ошибок в канале связи, при котором еще гарантируется секретность ключа. Принципиальный порог по вероятности ошибок составляет 43.75% (7/16) в режиме однофотонных входных состояний. Выяснен также вопрос о скорости генерации секретного ключа в реальном времени в зависимости от частотной полосы пропускания квантового канала связи. Кроме того, исследован вопрос о влиянии затухания в канале связи на секретность ключа. При наличии реального затухания в оптоволоконных линиях (0.2 db/km) приходится использовать входные многофотонные состояния для достижения приемлемых скоростей генерации ключа в реальном времени. Многофотонность входных состояний приближает их к классическому пределу и облегчает подслушивание. Выяснен вопрос о связи между вероятностью ошибки, вносимой подслушивателем, числом входных фотонов в сигнальных состояниях и длиной канала связи. Предложена также простая экспериментальная схема квантовой криптографии на базе оптоволоконного интерферометра Маха — Цандера, не требующая

поляризационного контроля и «идеальной» балансировки плеч интерферометра между входом и выходом.

О свойствах криптоалгоритма GI[2]

Б. А. Погорелов, М. А. Пудовкина

Одним из широко используемых поточных шифров является алгоритм поточного шифрования RC4, предложенный Р. Райвестом [1]. После того, как в 1993 году стало известно описание криптоалгоритма RC4, было предложено несколько поточные шифров, являющихся модификациями RC4 или основанными на общей идеи. Например, такими шифрами являются поточные шифры IA, IBAA, ISAAC, предложенные в [2].

В данной работе вводится алгоритм поточного шифрования GI, зависящей от параметров, конкретизируя которых получаем, в частности, криптоалгоритмы IA, IBAA, ISAAC.

Криптоалгоритм GI определяются шестью функциями $\varphi\colon \mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_{2^b}$, $\rho\colon \mathbb{Z}_m \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_m$, $\sigma\colon \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \to \mathbb{Z}_{2^b}$, $\delta\colon \mathbb{Z}_m \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_m$, $\chi\colon \mathbb{Z}_m \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_m$, $\varepsilon\colon \mathbb{N} \to \mathbb{N}$ ($\varepsilon\colon t \to t$ или $\varepsilon\colon t \to (t-1)$) и моделируются автономным автоматом $A_{\mathrm{GI}} = (\mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m, \mathbb{Z}_{2^b}, F_{\mathrm{GI}}, f_{\mathrm{GI}})$. Функции $F_{\mathrm{GI}}\colon \mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m$, $f_{\mathrm{GI}}\colon \mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_{2^b}$ будут описаны ниже. Криптоалгоритм GI также зависит от параметров $m = 2^n$, $b \geqslant 2n$, $n, b \in \mathbb{N}$.

Состоянием автомата A_{GI} в такте t ($t = 0, 1, \ldots$) является четверка $(i_t, a_t, q_t, s_t)\mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m$, где $s_t = \{s_t[0], \ldots, s_t[m-1]\}$ - таблица из m n-мерных двоичных векторов. Начальным состоянием является четверка $(0, a_0, q_0, s_0)$, причем a_0, q_0 предполагаются известными параметрами GI.

Приведем описание t-го ($t = 1, 2, \ldots$) такта работы A_{GI}.

Функция переходов F_{GI}:

$$\begin{aligned} i_t &= i_{t-1} + 1 \pmod m, \\ a_t &= \varphi(i_t, s_{t-1}, a_{t-1}), \\ s_t[i_t] &= (s_{t-1}[\rho(i_t, s_{t-1})] + \sigma(a_t, q_{t-1})) \pmod{2^b}, \\ s_t[k] &= s_{t-1}[k] \text{ при } k = \overline{0, m-1} \text{ и } k \neq i_t, \\ q_t &= (s_t[\delta(i_t, s_t)] + s_{\varepsilon(t)}[\chi(i_t, s_t)]) \pmod{2^b}. \end{aligned}$$

Функция выходов f_{GI}:

$$z_t = q_t.$$

Шифрование t-го знака открытого текста $x_t = (x_{t,b-1}, \ldots, x_{t,0}) \in \mathbb{Z}_2^b$ имеет вид $c_t = x_t \oplus z_t$. Расшифрование t-го знака шифртекста определяется выражением $x_t = c_t \oplus z_t$.

Криптоалгоритмы IA, IBAA, ISAAC, во введенной модели, имеют вид:

Криптоалгоритм IA

$\varepsilon\colon t \to (t-1)$, $\chi\colon (i, s) \to i$ $\delta\colon (i, s) \to (s[i] \gg n) \pmod m$, $\rho\colon (i, s) \to s[i] \pmod m$, $\sigma\colon (a, q) \to a + q \pmod{2^b}$, $\varphi(i, s, 0) \to 0$.

Криптоалгоритм IBAA

Пусть $p, q \in \mathbb{Z}_b$, $\omega\colon \mathbb{Z}_{2^b} \times \mathbb{Z}_b \times \mathbb{Z}_b \to \mathbb{Z}_{2^b}$ причем:

$$\omega(a, p, q) = (a \ll p) \oplus (a \gg q),$$

где $p + q = b$.

Тогда $\varepsilon\colon t \to (t-1)$, $\chi\colon (i, s) \to i$ $\delta\colon (i, s) \to (s[i] \gg n) \pmod m$, $\rho\colon (i, s) \to s[i] \pmod m$, $\sigma\colon (a, q) \to a + q \pmod{2^b}$, $\varphi(i, s, a) \to \omega(a, p, q) + s[(i + m/2) \pmod m] \pmod{2^b}$.

[2]Работа первого автора выполнена при поддержке гранта Президента РФ № НШ-2358.2003.9.

В [2] предложены следующие значения параметров: $p = 19$, $q = 13$.

Криптоалгоритм ISAAC

Пусть $p_0, p_1, p_3, p_4, \theta_1, \theta_2 \in \mathbb{Z}_b$, $G\colon \mathbb{Z}_{2^b}\colon \mathbb{N} \to \mathbb{Z}_{2^b}$, причем

$$G(a,t) = \begin{cases} ((a \ll p_0) \oplus a) & \text{при } t = 0 \pmod 4, \\ ((a \gg p_1) \oplus a) & \text{при } t = 1 \pmod 4, \\ ((a \ll p_2) \oplus a) & \text{при } t = 2 \pmod 4, \\ ((a \gg p_3) \oplus a) & \text{при } t = 3 \pmod 4. \end{cases}$$

Тогда$\chi\colon (i,s) \to i$, $\delta\colon (i,s) \to (s[i] \gg (n+\theta_2)) \pmod m$, $\sigma\colon (a,q) \to a+q \pmod{2^b}$, $\rho\colon (i,s) \to s[i] \gg \theta_1 \pmod m$, $\varepsilon\colon t \to (t-1)$, $\varphi(i,s,0) \to (G(a,t) + s[(i+m/2) \pmod m]) \pmod{2^b}$.

В [2] предложены следующие значения параметров: $p_0 = 13$, $\mathrm{p}_1 = 6$, $\mathrm{p}_2 = 2$, $\mathrm{p}_3 = 16$, $\theta_1 = \theta_2 = 2$.

Опишем некоторые теоретико-автоматные свойства автомата A_{GI}.

Будем обозначать через $s \pmod d$ таблицу $\{s[0] \pmod d, \ldots, s[m-1] \pmod d\}$.

Утверждение 1. *Пусть $s_0[i] = 0$, $i = \overline{0, m-1}$, или $s_0[i] = 2^{b-1}$, $i = \overline{0, m-1}$. Если для любого $i \in \mathbb{Z}_m$ справедливы равенства $\varphi(i, s_0, 0) = 0$, $\sigma(0,0) = 0$, $\rho(i, s_0) = 0$, $\delta(i, s_0) = 0$, $\chi(i, s_0) = 0$, то состояние $(0,0,0,s_0)$ принадлежит циклу длины m и соответствующая гамма нулевая.*

Утверждение 2. *Пусть существует такое натуральное число b_0, $n \leqslant b_0 < b$, что для любого состояния $(i,a,q,s) \in \mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m$ автомата A_{GI} выполняются равенства*

$$\begin{aligned} \varphi(i,s,a) &= \varphi(i, s \pmod{2^{b_0}}, a \pmod{2^{b_0}}) \pmod{2^{b_0}}, \\ \sigma(a,q) &= \sigma(a \pmod{2^{b_0}}, q \pmod{2^{b_0}}) \pmod{2^{b_0}}, \\ \rho(i,s) &= \rho(i, s \pmod{2^{b_0}}), \\ \delta(i,s) &= \delta(i, s \pmod{2^{b_0}}), \\ \chi(i,s) &= \chi(i, s \pmod{2^{b_0}}). \end{aligned}$$

Будем считать b_0 наименьшим таким числом. Тогда для любых состояний (i_0, a_0, q_0, s_0) и (i_0', a_0', q_0', s_0') таких, что $i_0 = i_0' \pmod m$, $q_0 = q_0' \pmod{2^{b_0}}$, $s_0[i] = s_0'[i] \pmod{2^{b_0}}$, $i = \overline{0, m-1}$, выполняются равенства $i_t = i_t'$, $a_t = a_t' \pmod{2^{b_0}}$, $s_t[i] = s_t'[i] \pmod{2^{b_0}}$, $i = \overline{0, m-1}$, $q_t = q_t' \pmod{2^{b_0}}$ для всех $t \geqslant 1$.

Следствие 1. *Пусть состояния $(i_0, 0, q_0, s_0)$ и $(i_0', 0, q_0', s_0')$ автомата A_{IA} такие, что $i_0 = i_0' \pmod m$, $q_0 = q_0' \pmod{m^2}$, $s_0[i] = s_0'[i] \pmod{m^2}$, $i = \overline{0, m-1}$. Тогда $i_t = i_t' \pmod m$, $q_t = q_t' \pmod{m^2}$, $s_t[i] = s_t'[i] \pmod{m^2}$ $i = \overline{0, m-1}$ для всех $t \geqslant 1$.*

Пусть функции $\varphi\colon \mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_{2^b}$, $\rho\colon \mathbb{Z}_m \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_m$, $\sigma\colon \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \to \mathbb{Z}_{2^b}$, $\delta\colon \mathbb{Z}_m \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_m$, $\chi\colon \mathbb{Z}_m \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_m$ такие, что выполняются условия утверждения 2. На множестве состояний A_{GI} введем бинарное отношение $\overline{b}_0$ следующим образом. Будем говорить, что состояния (i,a,q,s) и (i',a',q',s') связаны отношением $\overline{b}_0$ тогда и только тогда, когда $i = i'$, $a = a' \pmod{2^{b_0}}$, $q = q' \pmod{2^{b_0}}$, $s[i] = s'[i] \pmod{2^{b_0}}$, $i = \overline{0, m-1}$. Очевидно, что $\overline{b}_0$ есть отношение эквивалентности (b_0-эквивалентности).

Для формулировки следующего утверждения удобно обозначать $\mathrm{GI} = \mathrm{GI}(b)$, где b один из параметров криптоалгоритма GI.

Теорема 1. *Пусть выполняются условия утверждения 2. Тогда при $b0 \geqslant n$ автомат $A_{\mathrm{GI}(b_0)}$ есть гомоморфный образ автомата $A_{\mathrm{GI}(b)}$. Гомоморфизм задается парой сюръективных отображений ψ, υ, где*

$$\psi\colon \mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_m \times \mathbb{Z}_{2^{b_0}} \times \mathbb{Z}_{2^{b_0}} \times \mathbb{Z}_{2^{b_0}}^m, \quad \upsilon\colon \mathbb{Z}_{2^b} \to \mathbb{Z}_{2^{b_0}}$$

и

$$\begin{aligned} \psi(i,a,q,s) &= (i, a \pmod{2^{b_0}}, q \pmod{2^{b_0}}, s \pmod{2^{b_0}}), \\ \upsilon(z) &= z \pmod{2^{b_0}} \end{aligned}$$

для любых $z \in \mathbb{Z}_{2^b}$, $(i,a,q,s) \in \mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m$.

Следствие 2. *Пусть $b = 2n+\Delta$ ($\Delta > 0$). Тогда автомат $A_{\mathrm{IA(2n)}}$ есть гомоморфный образ автомата $A_{\mathrm{IA}(b)}$. Гомоморфизм задается парой сюръективных отображений ψ, υ, где*

$$\psi\colon \mathbb{Z}_m \times \{0\} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m \to \mathbb{Z}_m \times \{0\} \times \mathbb{Z}_{2^{2n}} \times \mathbb{Z}_{2^{2n}}^m, \quad \upsilon\colon \mathbb{Z}_{2^b} \to \mathbb{Z}_{2^{2n}}$$

и

$$\psi(i,0,q,s) = (i,0,q \pmod{2^{2n}}, s \pmod{2^{2n}}), \quad \upsilon(z) = z \pmod{2^{2n}}$$

для любых $z \in \mathbb{Z}_{2^b}$, $(i,0,q,s) \in \mathbb{Z}_m \times \{0\} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m$.

Утверждение 3. *Если в автомате A_{GI} функции φ и σ такие, что для любого состояния $(i,a,q,s) \in \mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m$ выполняются равенства $\varphi(i,s,a) = 0 \pmod 2$, $\sigma(a,q) = 0 \pmod 2$, $s_0[i] = 1 \pmod 2$, $i = \overline{0,m-1}$, $q_0 = 0 \pmod 2$, $a_0 = 0 \pmod 2$, то справедливы равенства $a_t = 0 \pmod 2$, $z_t = q_t = 0 \pmod 2$ и $s_t[i] = 1 \pmod 2$, $i = \overline{0,m-1}$, для всех $t \geqslant 1$.*

Утверждение 4. *Если в автомате A_{GI} функции φ и σ такие, что для некоторого натурального числа r, $1 \leqslant r \leqslant b$, и для любого состояния $(i,a,q,s) \in \mathbb{Z}_m \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b} \times \mathbb{Z}_{2^b}^m$ выполняются равенства $\varphi(i,s,a) = 0 \pmod{2^r}$, $\sigma(a,q) = 0 \pmod{2^r}$, $s_0[i] = 1 \pmod{2^r}$, $i = \overline{0,m-1}$, $q_0 = 0 \pmod{2^r}$, $a_0 = 0 \pmod{2^r}$, то справедливы равенства $s_t[i] = 0 \pmod{2^r}$, $i = \overline{0,m-1}$, $a_t = 0 \pmod{2^r}$, $q_t = z_t = 0 \pmod{2^r}$ для всех $t \geqslant 1$.*

Замечание. Легко проверить, что для автомата A_{IA} справедливы утверждения 3, 4.

Литература

[1] Rivest R. L. The RC4 encryption algorithm. RSA Data Security, Inc., Mar. 1992.

[2] Jenkins R. J., Jr. ISAAC. Fast Software Encryption, Cambridge 1996, vol. 1039, ed. by D. Gollmann, Springer-Verlag.

О новых конструкциях нелинейных фильтров для поточных шифраторов и их устойчивости против стандартных и новых криптографических атак

Ю. В. Таранников

Поточным шифратором, говоря немного упрощенно, как правило, называется устройство с памятью, которое после введения в него «ключа», определяющего начальные значения ячеек памяти, действует автономно и производит псевдослучайную последовательность, которая преобразует исходное сообщение побитово или побайтово в зашифрованное сообщение, например, складываясь с ним побитово по модулю 2. Главными требованиями к поточным шифраторам являются скорость их работы и надежность. Под надежностью понимается невозможность для противника за разумное время по некоторой имеющейся у него информации, например по схеме шифратора и перехваченным кускам выданной им псевдослучайной последовательности, определить всю псевдослучайную последовательность целиком, или, что равнозначно, раскрыть «ключ», что позволило бы противнику моментально читать все наши сообщения, зашифрованные с помощью этого ключа.

Одним из наиболее часто использующихся составных частей поточных шифраторов является Регистр Сдвига с Линейной Обратной Связью (РСЛОС), очень просто реализующийся как элемент микросхемы и очень быстро работающий. Однако использование одного только РСЛОС недостаточно, потому что существует много атак, позволяющих раскрывать «ключ» РСЛОС (начальные состояния его ячеек) за полиномиальное время относительно N — длины ключа, в то время как в идеале хотелось бы, чтобы противник не имел бы никакого более простого способа, чем перебирать все возможные варианты ключей, которых 2^N, и сравнивать производимые ими псевдослучайные последовательности

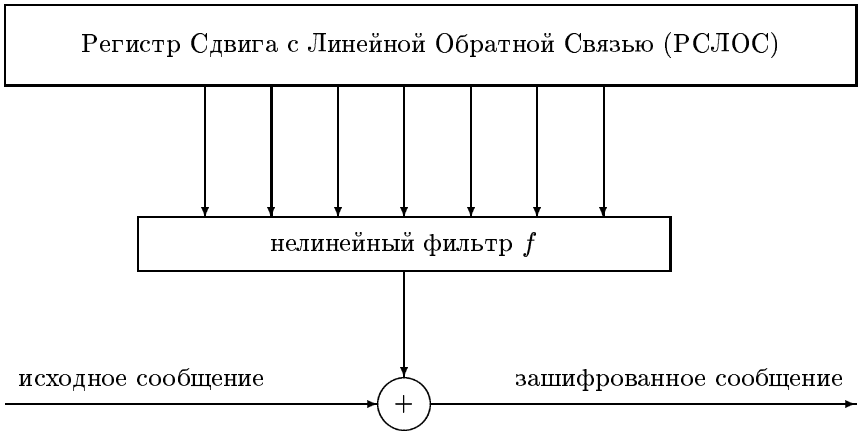


Рис. 1: Поточный шифратор, состоящий из РСЛОС и нелинейного фильтра.

с перехваченной. Т. е. не хотелось бы, чтобы существовала атака сложности меньше чем примерно 2^N операций. Поэтому для того, чтобы избавиться от линейной зависимости выдаваемой РСЛОС псевдослучайной последовательности от начальных состояний ячеек, значения некоторых n ячеек РСЛОС в каждый момент времени подают на нелинейный фильтр, представляющий собой булеву функцию от n переменных. И уже выходное значение булевой функции является очередным элементом псевдослучайной последовательности. Модель поточного шифратора, основанная на РСЛОС и нелинейном фильтре, показана на рис. 1. Существуют, конечно, и другие модели, однако указанная является одной из самых распространенных.

Естественно, что нелинейный фильтр на рис. 1 может быть выбран не как угодно. Существует много криптографических атак, которые позволяют противнику, если фильтр f выбран неподходящим образом, эффективно раскрыть ключ. Среди таких атак наиболее распространены корреляционные, линейные и другие. Поэтому был сформулирован ряд свойств, которым должны удовлетворять фильтры, чтобы успешно противостоять таким атакам. Наиболее важными свойствами являются уравновешенность, высокие нелинейность, алгебраическая степень, корреляционная иммунность. Кроме того, конечно, для практического использования и быстродействия шифратора фильтр должен иметь простую реализацию. Перечисленные свойства часто противоречат друг другу, о чем свидетельствуют и теоретические результаты. Так, неравенство, связывающее нелинейность и корреляционную иммунность, было установлено докладчиком в [5]. Чтобы найти фильтры, удовлетворяющие требуемым параметрам криптографической надежности, есть два пути. Один состоит в компьютерном поиске требуемых фильтров и применим только в случае, если число входов фильтра относительно мало (где-то не более 8–10 входов для полного перебора и 12–14 для эвристического поиска). Второй подход состоит в построении алгебраических конструкций и теоретическом доказательстве того, что построенные с их помощью фильтры удовлетворяют заданным параметрам. Тут надо иметь в виду и то, что сложность реализации такой конструкции должна быть практически приемлемой. Ведь сложность реализации типичной булевой функции от n переменных является экспоненциальной по n.

Известно несколько семейств фильтров на основе упомянутых алгебраических конструкций. Среди них и семейство, строящееся с помощью рекурсивных конструкций, разработанных и развитых докладчиком в работах [1, 4, 5, 6, 7]. В частности на фильтрах, задаваемых этими конструкциями, для широких границ параметров были впервые достигнуты теоретически оптимальные соотношения между нелинейностью и корреляционной иммунностью, а также алгебраической степенью каждого входа. Конструкции позволяют строить фильтры со сколь угодно большим числом входов с линейной по числу входов сложностью. Общая схема поточного шифратора, состоящего из РСЛОС и нелинейного фильтра, построенного на основе разработанной докладчиком конструкции, показана на рис. 2. Общая схема блока Б проста для практического использования, но слишком сложна для воспроизведения ее в настоящем докладе. С подробным изложением конструкций докладчика и сопутствующей теорией можно ознакомиться в работе [1], размещенной также на сайте `www.cryptography.ru`.

В 2002–2003 годах французским исследователем Николасом Куртуа (N. Courtois) и рядом его коллег был разработан новый вид криптографической атаки — так называемая алгебраическая атака [2, 3], оказавшаяся весьма эффективной против рассматриваемого нами здесь типа поточных шифраторов.

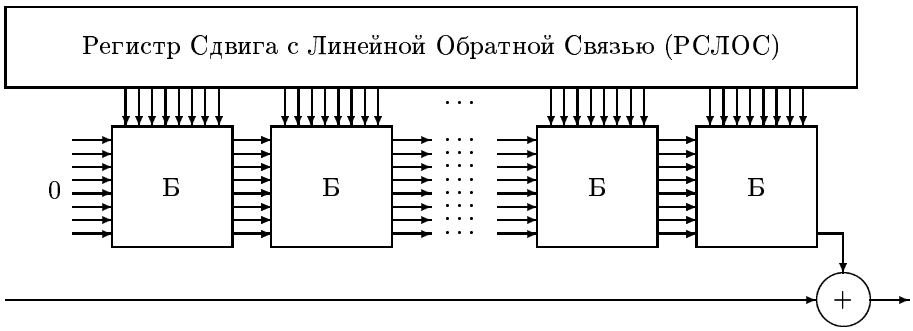


Рис. 2: Схема поточного шифратора, состоящего из РСЛОС и нелинейного фильтра, построенного с помощью нашей рекурсивной конструкции.

Применение этой атаки основано на построении точной или приближенной (в случае нахождения очень хорошей аппроксимации фильтра) системы нелинейных уравнений невысокой степени, связывающей начальные значения ячеек памяти РСЛОС и значения перехваченной противником последовательности, и решения этой системы путем последующей линеаризации и получения из нее переопределенной системы линейных уравнений. В связи с разработкой этой атаки Н. Куртуа выдвинул новые требования надежности, которым должна удовлетворять булева функция f, используемая в качестве нелинейного фильтра: функция f не только не должна иметь хорошей аппроксимации функциями невысокой степени, но и не должно существовать функции g невысокой степени, такой что функция $f \cdot g$ тоже невысокой степени или хорошо аппроксимируется функцией невысокой степени. Н. Куртуа показал, что против алгебраической атаки любой шифратор с фильтром, у которого не более чем десять входов, заведомо не удовлетворяет требуемым критериям надежности. Что касается большинства известных алгебраических конструкций фильтров с большим числом входов, то для них также были предложены алгебраические атаки, намного более эффективные, чем простой перебор ключей.

Иначе обстоит дело с рекурсивными конструкциями, разработанными докладчиком. Предварительные исследования, проведенные докладчиком и его учениками, показали, что для многих последовательностей фильтров, строящихся с помощью этой конструкции, максимальная степень «нехорошей» функции g, упомянутой выше, растет с ростом числа итераций (числа блоков Б на рис. 2), и, таким образом, эти фильтры претендуют на то, чтобы быть устойчивыми относительно алгебраической атаки. Однако еще предстоит много работы по точной оценке роста степени «нехорошей» функции g в зависимости от параметров конструкции, нахождению наиболее оптимальных таких параметров в смысле надежности и быстродействия шифратора, предъявлению рекомендаций по количеству числа итераций (блоков Б), достаточному для достижения требуемой надежности, экономной схемной и компьютерной реализации шифратора для увеличения быстродействия. Продолжение исследований в этих направлениях представляется очень перспективным.

Литература

[1] Таранников Ю. В. О корреляционно-иммунных и устойчивых булевых функциях. Математические вопросы кибернетики. Вып. 11, М.: Физматлит, 2002, с. 91–148.

[2] Courtois N. Higher order correlation attacks, XL algorithm, and cryptanalysis of Toyocrypt. Proceedings of 5th International Conference on Information Security and Cryptology (ICISC 2002), November 28–29, 2002, Seoul, Korea. Lecture Notes in Computer Science, v. 2587, p. 182–199. Springer-Verlag, 2002.

[3] Courtois N., Meier W. Algebraic attacks on stream ciphers with linear feedback. Advanced in Cryptology: Eurocrypt 2003, Warsaw, Poland, May 4–8, 2003, Proceedings. Lecture Notes in Computer Science, v. 2656, p. 345–357. Springer-Verlag, 2003.

[4] Fedorova M., Tarannikov Yu. On the constructing of highly nonlinear resilient Boolean functions by means of special matrices. Progress in Cryptology — Indocrypt 2001, Chennai, India, December 16–20, 2001, Proceedings. Lecture Notes in Computer Science, v. 2247, p. 254–266. Springer-Verlag, 2001.

[5] Tarannikov Yu. On resilient Boolean functions with maximal possible nonlinearity. Proceedings of Indocrypt 2000, Calcutta, India, December 10–13, 2000. Lecture Notes in Computer Science, v. 1977, p. 19–30. Springer-Verlag, 2000.

[6] Tarannikov Yu. New constructions of resilient Boolean functions with maximal nonlinearity. Fast Software Encryption. 8th International Workshop, FSE 2001 Yokohama, Japan, April 2-4, 2001. Revised Papers. Lecture Notes in Computer Science, v. 2355, 2002, p. 66–77.

[7] Tarannikov Yu., Korolev P., Botev A. Autocorrelation coefficients and correlation immunity of Boolean functions. Proceedings of Asiacrypt 2001, Gold Coast, Australia, December 9–13, 2001. Lecture Notes in Computer Science, v. 2248, p. 460–479. Springer-Verlag, 2001.

Корреляционная иммунность и реальная секретность[3]

О. А. Логачёв, А. А. Сальников, В. В. Ященко

Рассмотрим комбинирующий генератор (см. рис. 1), построенный с помощью n регистров сдвига с линейными обратными связями — LFSR-i, $i = 1, 2, \ldots, n$. Длины регистров будем обозначать k_i, $i = 1, 2, \ldots, n$ соответственно. Шифрующая последовательность получается с помощью комбинирующей булевой функции $f\left(x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(n)}\right) = f(\boldsymbol{x})$от n переменных.

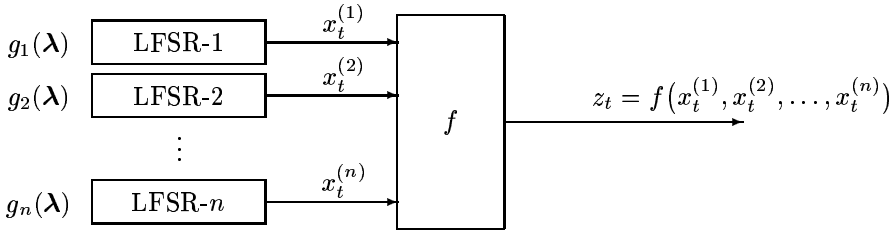


Рис. 1:

Будем считать, что полиномы обратных связей регистров сдвига $g_i(\boldsymbol{\lambda})$, $i = 1, 2, \ldots, n$ примитивны и регистры порождают линейные рекуррентные последовательности максимального периода $2^{k_i} - 1$, $i = 1, 2, \ldots, n$.

Обозначим для натурального числа N и фиксированного i, $i = 1, 2, \ldots, n$, $\boldsymbol{x}^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_N^{(i)}\right) = \left\{ x_t^{(i)} \right\}_{t=1}^{N}$ — последовательность длины N, вырабатываемую регистром сдвига с номером i, находившимся в начальном состоянии $\left(x_1^{(i)}, x_2^{(i)}, \ldots, x_{k_i}^{(i)}\right)$. То есть, в такт t регистр с номером i вырабатывает элемент $x_t^{(i)}$ последовательности $\boldsymbol{x}^{(i)}$ и комбинирующий генератор вырабатывает знак $z_t = f\left(x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(n)}\right)$ шифрующей последовательности $\boldsymbol{z} = \left\{ z_t \right\}_{t=1}^{N}$.

Пусть известна последовательность $\boldsymbol{z}$, выработанная комбинирующим генератором и задача состоит в определении начальных состояний $\left(x_1^{(1)}, x_2^{(1)}, \ldots, x_{k_1}^{(1)}\right)$, $\left(x_1^{(2)}, x_2^{(2)}, \ldots, x_{k_2}^{(2)}\right)$, $\ldots$, $\left(x_1^{(n)}, x_2^{(n)}, \ldots, x_{k_n}^{(n)}\right)$ регистров, при которых была получена последовательность $\boldsymbol{z}$. Другими словами, задача состоит в восстановлении последовательностей $\boldsymbol{x}^{(1)} = \left\{ x_t^{(1)} \right\}_{t=1}^{N}$, $\boldsymbol{x}^{(2)} = \left\{ x_t^{(2)} \right\}_{t=1}^{N}$, $\ldots$, $\boldsymbol{x}^{(n)} = \left\{ x_t^{(n)} \right\}_{t=1}^{N}$.

[3]Работа поддержана Российским фондом фундаментальных исследований (номера проектов 02-01-00581 и 02-01-00687).

Эту задачу решают с помощью различных вариантов корреляционного метода: с помощью статистических процедур [4], на основе теоретико-кодового подхода [2, 3] и т.д. В результате анализа этой задачи в работах [4, 5] была выдвинута концепция корреляционно-иммунных булевых функций.

Пусть $X^{(1)}, X^{(2)}, \ldots, X^{(n)}$ — независимые двоичные одинаково распределенные случайные величины, $P\{ X^{(i)} = 0 \} = P\{ X^{(i)} = 1 \} = \dfrac{1}{2}$, $i = 1, 2, \ldots, n$.

Определение 1 ([4]). Булева функция $f\left(x^{(1)}, \ldots, x^{(n)}\right)$ от n переменных называется *корреляционно-иммунной порядка* m, если для любого набора $1 \leqslant i_1 < \ldots < i_m \leqslant n$ равна нулю взаимная информация $I\Big(\left(X^{(i_1)}, X^{(i_2)}, \ldots, X^{(i_m)}\right), Z \Big)$, где случайная величина Z определена равенством: $Z = f\left(X^{(1)}, \ldots, X^{(n)}\right)$.

Уравновешенная корреляционно-иммунная порядка m функция называется *m-устойчивой*.

Криптографические приложения связаны с устойчивыми функциями.

Основным следствием определения 1 является то, что при нахождении по последовательности $\boldsymbol{z} = \{ z_t \}_{t=1}^{N}$ начальных состояний комбинирующего генератора корреляционным методом, на первом этапе мы должны определять начальные состояния не менее, чем $m + 1$ регистра сдвига.

В настоящей работе показано, что существуют другие методы (отличные от корреляционного), позволяющие в данном случае определять на первом этапе начальное состояние не более, чем m регистров сдвига.

Обозначим $\mathbb{F}_2$ — конечное поле из двух элементов, V_l — пространство векторов длины l над этим полем, для вектора $\boldsymbol{u} \in V_l$ обозначим $\mathrm{wt}(\boldsymbol{u})$ — его вес Хэмминга.

Пусть p и r — натуральные числа такие, что $n \geqslant r > p \geqslant 0$, отображение $\Phi : V_s \to V_r$ удовлетворяет условию $\mathrm{wt}(\Phi(\boldsymbol{u})) \geqslant p$ для любого $\boldsymbol{u} \in V_s$, g — булева функция от s переменных.

Будем считать, что функция f принадлежит классу Майорана — МакФарланда. Воспользовавшись координатным представлением отображения $\Phi = \left(\Phi_1, \Phi_2, \ldots, \Phi_r \right)$, представим функцию в виде

$$f(\boldsymbol{x}) = f_{\Phi,g}(\boldsymbol{x}) = \bigoplus_{i=1}^{r} x^{(i)} \Phi_i \left(x^{(r+1)}, \ldots, x^{(n)} \right) \oplus g \left(x^{(r+1)}, \ldots, x^{(n)} \right). \tag{1}$$

Будем также считать, что $s = n - r \leqslant m$. Ясно, что функция f — является m-устойчивой функцией при $m \geqslant p$ (см. [5]).

Используя корреляционный метод для нахождения начальных состояний комбинирующего генератора с функцией (1), необходимо на первом этапе восстанавливать начальные состояния не менее, чем $m + 1$ регистров.

Рассмотрим следующий алгоритм восстановления начальных состояний комбинирующего генератора с функцией (1) по известной последовательности $\boldsymbol{z}$. Будем опробовать начальные состояния регистров сдвига с номерами $r + 1, r + 2, \ldots, n$, а именно, $\left(x_1^{(r+1)}, x_2^{(r+1)}, \ldots, x_{k_{r+1}}^{(r+1)} \right)$, $\left(x_1^{(r+2)}, x_2^{(r+2)}, \ldots, x_{k_{r+2}}^{(r+2)} \right)$, $\ldots$, $\left(x_1^{(n)}, x_2^{(n)}, \ldots, x_{k_n}^{(n)} \right)$, вырабатывая последовательности $\boldsymbol{x}^{(r+1)}$, $\boldsymbol{x}^{(r+2)}$, $\boldsymbol{x}^{(n)}$.

Пусть $\left(b_t^{(r+1)}, b_t^{(r+2)}, \ldots, b_t^{(n)} \right) \in V_s$, $t = 1, \ldots, N$ — известные нам последовательности соответствующих регистров. Воспользовавшись равенством (1), получим линейную систему уравнений относительно оставшихся переменных

$$\begin{cases} z_t = \bigoplus\limits_{i=1}^{r} x^{(i)} \Phi_i \left(b_t^{(r+1)}, \ldots, b_t^{(n)} \right) \oplus g \left(b_t^{(r+1)}, \ldots, b_t^{(n)} \right), \\ t = 1, 2, \ldots, N. \end{cases} \tag{2}$$

Рассмотрим такие такты $t_1, t_2, \ldots, t_M$, для которых $\Phi \left(b_{t_\nu}^{(r+1)}, \ldots, b_{t_\nu}^{(n)} \right) = (\underbrace{1, \ldots, 1}_{p}, 0, \ldots, 0)$, $\nu = 1, \ldots, M$ (будем считать, что такие вектора лежат в образе отображения Φ). Если считать, что отображение Φ «ведёт» себя как случайное, то в среднем таких тактов будет не менее, чем $N / \left(2^n - \sum\limits_{q=0}^{p-1} \binom{n}{q}\right)$.

Обозначив $z'_t = z_t \oplus g\left(b_t^{(r+1)}, \ldots, b_t^{(n)}\right)$, приведём систему (2) к виду

$$\begin{cases} x_{t_1}^{(1)} \oplus \ldots \oplus x_{t_1}^{(p)} = z'_{t_1}, \\ \ldots \\ x_{t_M}^{(1)} \oplus \ldots \oplus x_{t_M}^{(p)} = z'_{t_M}. \end{cases} \tag{3}$$

Поскольку для каждой линейной рекуррентной последовательности $\boldsymbol{x}^{(i)}$, $i = 1, 2, \ldots, n$, каждый её элемент $x_t^{(i)}$ является линейной комбинацией её первых членов $x_1^{(i)}, x_2^{(i)}, \ldots, x_{k_i}^{(i)}$, то его можно представить как $x_t^{(i)} = c_{t,1}^{(i)} x_1^{(i)} \oplus c_{t,2}^{(i)} x_2^{(i)} \oplus \ldots \oplus c_{t,k_i}^{(i)} x_{k_i}^{(i)}$, где $c_{t,1}^{(i)}, \ldots, c_{t,k_i}^{(i)} \in \mathbb{F}_2$ — константы, определяемые многочленом обратной связи $g_i(\boldsymbol{\lambda})$. Тогда система (3) принимает вид

$$\begin{cases} c_{t_1,1}^{(1)} x_1^{(1)} \oplus \ldots \oplus c_{t_1,k_1}^{(1)} x_{k_1}^{(1)} \oplus \ldots \oplus c_{t_1,1}^{(p)} x_1^{(p)} \oplus \ldots \oplus c_{t_1,k_p}^{(p)} x_{k_p}^{(p)} = z'_{t_1}, \\ \ldots \\ c_{t_M,1}^{(1)} x_1^{(1)} \oplus \ldots \oplus c_{t_M,k_1}^{(1)} x_{k_1}^{(1)} \oplus \ldots \oplus c_{t_M,1}^{(p)} x_1^{(p)} \oplus \ldots \oplus c_{t_M,k_p}^{(p)} x_{k_p}^{(p)} = z'_{t_M}, \end{cases} \tag{4}$$

то есть представляет собой линейную систему из M уравнений относительно $k_1 + k_2 + \ldots + k_p$ неизвестных.

Если мы правильно угадали начальное состояние $\left(\text{и}_1^{(j)}, b_2^{(j)}, \ldots, b_{k_j}^{(j)}\right) \in V_s$, $j = r+1, \ldots, n$, то система (4) совместна. Если же мы не угадали начальное состояние, то система (4) может быть как совместна, так и несовместна. В случае её несовместности можно отбраковывать начальные состояния регистров с номерами $r+1, r+2, \ldots, n$. Для того, чтобы оценить вероятность несовместности системы (4), введём следующую статистическую модель. Будем предполагать, что величины $c_{t_\nu,i}^{(j)}$, $j = 1, 2, \ldots, p$, $\nu = 1, 2, \ldots, M$, $i = 1, 2, \ldots, k_j$ являются независимыми в совокупности одинаково распределёнными двоичными случайными величинами, $P\{\, c_{t_\nu,i}^{(j)} = 0 \,\} = P\{\, c_{t_\nu,i}^{(j)} = 1 \,\} = \dfrac{1}{2}$.

Пусть

$$A = \begin{pmatrix} c_{t_1,1}^{(1)}, & \cdots, & c_{t_1,k_1}^{(1)}, & \cdots, & c_{t_1,1}^{(p)}, & \cdots, & c_{t_1,k_p}^{(p)} \\ \multicolumn{7}{c}{\dotfill} \\ c_{t_M,1}^{(1)}, & \cdots, & c_{t_M,k_1}^{(1)}, & \cdots, & c_{t_M,1}^{(p)}, & \cdots, & c_{t_M,k_p}^{(p)} \end{pmatrix} \quad \text{и}$$

$$A' = \begin{pmatrix} & \vdots & z'_{t_1} \\ A & \vdots & \vdots \\ & \vdots & z'_{t_M} \end{pmatrix}.$$

Необходимым и достаточным условием несовместности системы (4) является неравенство

$$\operatorname{rank} A \neq \operatorname{rank} A'.$$

Для вероятности несовместности системы (4) $p_{\text{н}}$ справедливо неравенство

$$p_{\text{н}} \geqslant p_{\text{н},k+1} = P\{\operatorname{rank} A' = k+1\},$$

где $k = k_1 + k_2 + \ldots + k_p$. Вычислим вероятность $p_{\text{н},k+1}$.

При фиксации начальных заполнений регистров с номерами $r+1, r+2, \ldots, n$ столбец $\left(z'_{t_1}, \ldots, z'_{t_M}\right)^T$ матрицы A' определён. Будем считать, что он не нулевой (выбором параметра M можно вероятность $P\{\left(z'_{t_1}, \ldots, z'_{t_M}\right)^T \neq (0, \ldots, 0)^T\}$ сколь угодно близко приблизить к 1). Для того, чтобы ранг матрицы A' был полный, второй столбец матрицы A' должен быть линейно независим от $\left(z'_{t_1}, \ldots, z'_{t_M}\right)^T$, т.е. он может быть выбран одним из $2^M - 2$ способов; далее, для того, чтобы ранг матрицы A' был полный, третий столбец матрицы A' должен быть линейно независим от первых двух, т.е. он может быть выбран одним из $2^M - 2^2$ способами и т.д. Наконец k-ый столбец может быть выбран одним из

$2^M - 2^k$ способом (при этом мы считаем, что $M > k$). Таким образом

$$p_{\text{н},k+1} = \frac{2^M - 2}{2^m} \cdot \frac{2^M - 2^2}{2^m} \cdot \ldots \cdot \frac{2^M - 2^k}{2^m} = \prod_{i=1}^{k} \left(1 - \frac{2^i}{2^M}\right) =$$

$$= \prod_{i=1}^{k} \left(1 - \frac{1}{2^{M-i}}\right) \geqslant \left(1 - \frac{1}{2^{M-k}}\right)^k.$$

Будем теперь M выбирать кратным k, то есть $M = uk$, $u > 1$ (это упрощает выкладки, но необходимая оценка может быть получена и для произвольного M). Имеем

$$p_{\text{н},k+1} \geqslant \left(1 - \frac{1}{2^{(u-1)k}}\right)^k$$

и

$$1 - p_{\text{н},k+1} \leqslant 1 - \left(1 - \frac{1}{2^{(u-1)k}}\right)^k = \binom{k}{1}\frac{1}{\left(2^{u-1}\right)^k} - \binom{k}{2}\frac{1}{\left(2^{u-1}\right)^{2k}} + \ldots \leqslant$$

$$\leqslant \frac{k}{2^{(u-1)k}}.$$

Следовательно, вероятность $p_{\text{н},k+1}$ выбором параметра u может быть сделана сколь угодно близкой к 1. Поэтому, неверно угадав начальные состояния регистров сдвига с номерами $r+1, r+2, \ldots, n$, мы получим несовместную систему (4) линейных уравнений и отбракуем эти начальные состояния.

Вычисление ранга матрицы A' можно проводить методом Гаусса. Это потребует $\mathcal{O}(M)$ элементарных операций.

Таким образом, уже на первом шаге алгоритма будут найдены начальные состояния p регистров, $p \leqslant m$, что невозможно осуществить, пользуясь корреляционным методом.

Литература

[1] Carlet C. A Large Class of Cryptographic Boolean Functions via a Study of the Maiorana–McFarland Constructions // Advances in Cryptology: CRYPTO'02/ Lect. Notes in Comput. Sci. — **Vol. 2442.** — New York: Springer-Verlag. — 2002. — P. 549–564.

[2] Chepyzhov V., Smeets B. On a Fast Correlation Attacks on Certain Stream Ciphers. Advances in Cryptology: EUROCRYPT'91 // Lect. Notes in Comput. Sci. — **Vol. 547.** — New York: Springer-Verlag. — 1991. — P. 176–185.

[3] Meier W., Staffelbach O. Fast Correlation Attacks on Certain Stream Ciphers // Journal of Cryptology. — **Vol. 1.** — No. 3. — P. 159–176. — 1989.

[4] Siegentaler T. Correlation-immunity of Nonlinear Combining Functions for Cryptographic Applications. // IEEE Trans. on Information Theory. — 1984. — **Vol. IT-30.** — 5. — P. 776–780.

[5] Siegentaler T. Decrypting a Class of Stream Ciphers Using Ciphertext Only. // IEEE Trans. on Computers. — 1985. — **Vol. C-34.** — 1. — P. 81–85.

Аппроксимация булевых функций элементами биортогонального базиса[4]

О. А. Логачёв, А. А. Сальников, В. В. Ященко

Систему из 2^n булевых функций $e_{\boldsymbol{\alpha}}(\boldsymbol{x})$, $\boldsymbol{\alpha}, \boldsymbol{x} \in \mathbb{F}_2^n$ будем называть *биортогональным базисом*, если

[4]Работа поддержана Российским фондом фундаментальных исследований (номера проектов 02-01-00581 и 02-01-00687).

выполнены условия

$$\sum_{\boldsymbol{x}\in\mathbb{F}_2^n}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}(-1)^{e_{\boldsymbol{\beta}}(\boldsymbol{x})}=0, \qquad \text{если } \boldsymbol{\alpha}\neq\boldsymbol{\beta}, \tag{1}$$

$$\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{y})}=0, \qquad \text{если } \boldsymbol{x}\neq\boldsymbol{y}. \tag{2}$$

Это означает, что система функций $e_{\boldsymbol{\alpha}}(\boldsymbol{x})$ задает два ортогональных базиса в 2^n-мерном действительном векторном пространстве. Векторы первого базиса — это

$$\Big((-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x}_1)},\dots,(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x}_{2^n})}\Big), \qquad \text{для всех } \boldsymbol{\alpha}\in\mathbb{F}_2^n,$$

векторы второго базиса — это

$$\Big((-1)^{e_{\boldsymbol{\alpha}_1}(\boldsymbol{x})},\dots,(-1)^{e_{\boldsymbol{\alpha}_{2^n}}(\boldsymbol{x})}\Big), \qquad \text{для всех } \boldsymbol{x}\in\mathbb{F}_2^n;$$

здесь $\{\,\boldsymbol{x}_1,\dots,\boldsymbol{x}_{2^n}\,\}$, $\{\,\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_{2^n}\,\}$ — две произвольных фиксированных нумерации элементов пространства $\mathbb{F}_2^n$. На языке матриц условия (1), (2) означают, что матрица $\big\|(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}\big\|_{\boldsymbol{\alpha},\boldsymbol{x}\in\mathbb{F}_2^n}$ размера $2^n\times 2^n$ является матрицей Адамара по строкам (условие (1)) и по столбцам (условие (2)). Из теории векторных пространств хорошо известно, что любой вектор однозначно разлагается по ортогональному базису, причем коэффициенты разложения легко выражаются с помощью операции скалярного произведения. Применительно к интересующему нас случаю булевых функций это разложение и выражения для коэффициентов выглядят следующим образом

$$\breve{f}(\boldsymbol{\alpha})=\sum_{\boldsymbol{x}\in\mathbb{F}_2^n}(-1)^{f(\boldsymbol{x})}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}, \qquad \text{для всех } \boldsymbol{\alpha}\in\mathbb{F}_2^n, \tag{3}$$

$$(-1)^{f(\boldsymbol{x})}=\frac{1}{2^n}\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\breve{f}(\boldsymbol{\alpha})(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}, \qquad \text{для всех } \boldsymbol{x}\in\mathbb{F}_2^n. \tag{4}$$

Обозначение $\breve{f}(\boldsymbol{\alpha})$ и нормирующий коэффициент выбраны так, чтобы в случае $e_{\boldsymbol{\alpha}}(\boldsymbol{x})=\langle\boldsymbol{\alpha},\boldsymbol{x}\rangle$ получить $\breve{f}(\boldsymbol{\alpha})=\widehat{f}(\boldsymbol{\alpha})$ и возвратиться к преобразованию Уолша — Адамара. Кроме модельного простейшего случая $e_{\boldsymbol{\alpha}}(\boldsymbol{x})=\langle\boldsymbol{\alpha},\boldsymbol{x}\rangle$ всюду ниже будем иметь в виду следующий важный класс биортогональных базисов.

Пример 1. Пусть $n=k+2l$ и поэтому любой вектор $\boldsymbol{x}\in\mathbb{F}_2^n$ можно представить в виде $\boldsymbol{x}=\big(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)}\big)$, где $\boldsymbol{x}^{(1)}\in\mathbb{F}_2^k$, $\boldsymbol{x}^{(2)}\in\mathbb{F}_2^{2l}$. Пусть $\varphi\big(\boldsymbol{x}^{(2)}\big)$ — некоторая бент-функция. Положим

$$e_{\boldsymbol{\alpha}}(\boldsymbol{x})=e_{(\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)})}\Big(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)}\Big)=\langle\boldsymbol{\alpha}^{(1)},\boldsymbol{x}^{(1)}\rangle\oplus\varphi\big(\boldsymbol{x}^{(2)}\oplus\boldsymbol{\alpha}^{(2)}\big).$$

Система $e_{\boldsymbol{\alpha}}(\boldsymbol{x})$ является биортогональным базисом. Условия (1), (2) легко проверяются с учетом того, что $\varphi\big(\boldsymbol{x}^{(2)}\big)$ — бент-функция:

$$\sum_{\boldsymbol{x}\in\mathbb{F}_2^n}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}(-1)^{e_{\boldsymbol{\beta}}(\boldsymbol{x})}=$$

$$=\sum_{\boldsymbol{x}^{(1)}\in\mathbb{F}_2^k}(-1)^{\langle\boldsymbol{\alpha}^{(1)}\oplus\boldsymbol{\beta}^{(1)},\boldsymbol{x}^{(1)}\rangle}\cdot\sum_{\boldsymbol{x}^{(2)}\in\mathbb{F}_2^{2l}}(-1)^{\varphi(\boldsymbol{\alpha}^{(2)}\oplus\boldsymbol{x}^{(2)})\oplus\varphi(\boldsymbol{\beta}^{(2)}\oplus\boldsymbol{x}^{(2)})}=0,$$

если $\big(\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}\big)\neq\big(\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)}\big)$,

$$\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{y})}=$$

$$=\sum_{\boldsymbol{\alpha}^{(1)}\in\mathbb{F}_2^k}(-1)^{\langle\boldsymbol{x}^{(1)}\oplus\boldsymbol{y}^{(1)},\boldsymbol{\alpha}^{(1)}\rangle}\cdot\sum_{\boldsymbol{\alpha}^{(2)}\in\mathbb{F}_2^{2l}}(-1)^{\varphi(\boldsymbol{\alpha}^{(2)}\oplus\boldsymbol{x}^{(2)})\oplus\varphi(\boldsymbol{\alpha}^{(2)}\oplus\boldsymbol{y}^{(2)})}=0,$$

если $\big(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)}\big)\neq\big(\boldsymbol{y}^{(1)},\boldsymbol{y}^{(2)}\big)$.

Заметим теперь, что на языке теории кодирования из равенства (3) легко получить выражение для расстояния Хэмминга от вектора значений булевой функции $f(\boldsymbol{x})$ до вектора значений булевой функции $e_{\boldsymbol{\alpha}}(\boldsymbol{x})$:

$$\rho\left(f(\boldsymbol{x}), e_{\boldsymbol{\alpha}}(\boldsymbol{x})\right) = 2^{n-1} - \frac{1}{2}\breve{f}(\boldsymbol{\alpha}). \tag{5}$$

Отсюда, в частности, следует, что ближе всего (в смысле расстояния Хэмминга) к функции $f(\boldsymbol{x})$ из системы $e_{\boldsymbol{\alpha}}(\boldsymbol{x})$ лежит функция $e_{\boldsymbol{\alpha}_0}(\boldsymbol{x})$ с тем номером $\boldsymbol{\alpha}_0 \in \mathbb{F}_2^n$, для которого число $\breve{f}(\boldsymbol{\alpha}_0)$ максимально среди всех $\breve{f}(\boldsymbol{\alpha})$. Тем самым задача аппроксимации булевых функций элементами биортогонального базиса сведена к задаче нахождения $\max\limits_{\boldsymbol{\alpha}\in\mathbb{F}_2^n} \breve{f}(\boldsymbol{\alpha})$.

Чтобы ещё больше подчеркнуть преемственность рассматриваемого общего случая биортогонального базиса по отношению к модельному случаю $e_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \langle \boldsymbol{\alpha}, \boldsymbol{x}\rangle$, некоторые результаты удобно изложить на языке теории кодирования. Для этого введем ещё одно новое понятие.

Кодом Адамара $H\left(e_{\boldsymbol{\alpha}}(\boldsymbol{x})\right)$ назовём множество строк, которые являются векторами значений булевых функций $e_{\boldsymbol{\alpha}}(\boldsymbol{x})$, $e_{\boldsymbol{\alpha}}(\boldsymbol{x}) \oplus 1$, $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, где $e_{\boldsymbol{\alpha}}(\boldsymbol{x})$ — некоторый биортогональный базис. Код Адамара, вообще говоря, нелинейный, кроме случая $e_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \langle \boldsymbol{\alpha}, \boldsymbol{x}\rangle$, когда код Адамара совпадает с кодом Рида — Маллера первого порядка. Значения основных параметров кода Адамара сведём в одно утверждение.

Теорема 1. *Произвольный код Адамара имеет следующие параметры:*

а) длина — 2^n;

б) мощность — 2^{n+1};

в) кодовое расстояние — 2^{n-1};

г) радиус покрытия — $\leqslant 2^{n-1} - 2^{\frac{n}{2}-1}$.

Доказательство. Утверждения а) и б) очевидны, утверждение в) вытекает из (1). Для доказательства утверждения г) докажем для чисел $\breve{f}(\boldsymbol{\alpha})$, где $f(\boldsymbol{\alpha})$ — произвольная булева функция, аналог равенства Парсеваля. Из (3) с учетом (2) имеем

$$\sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n} \left(\breve{f}(\boldsymbol{\alpha})\right)^2 = \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n} \sum_{\boldsymbol{x},\boldsymbol{y}\in\mathbb{F}_2^n} (-1)^{f(\boldsymbol{x})}(-1)^{f(\boldsymbol{y})}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{y})} =$$

$$= \sum_{\boldsymbol{x},\boldsymbol{y}\in\mathbb{F}_2^n} (-1)^{f(\boldsymbol{x})}(-1)^{f(\boldsymbol{y})} \sum_{\boldsymbol{\alpha}\in\mathbb{F}_2^n} (-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{y})} = 2^{2n}. \tag{6}$$

Отсюда

$$\min_{f} \max_{\boldsymbol{\alpha}\in\mathbb{F}_2^n} \left|\breve{f}(\boldsymbol{\alpha})\right| \leqslant 2^{n/2} \tag{7}$$

С учетом (5) мы приходим к утверждению г). □

Теорема 2. *Если* $n = 2m + 2l$ *и* $\varphi\left(\boldsymbol{x}^{(2)}\right)$ *— некоторая бент-функция,*

$$e_{\boldsymbol{\alpha}}(\boldsymbol{x}) = e_{(\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)})}\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right) = \langle \boldsymbol{\alpha}^{(1)}, \boldsymbol{x}^{(1)}\rangle \oplus \varphi\left(\boldsymbol{x}^{(2)} \oplus \boldsymbol{\alpha}^{(2)}\right),$$

$\boldsymbol{\alpha}^{(1)}, \boldsymbol{x}^{(1)} \in \mathbb{F}_2^{2m}$, $\boldsymbol{\alpha}^{(2)}, \boldsymbol{x}^{(2)} \in \mathbb{F}_2^{2l}$, *то радиус покрытия кода Адамара* $H\left(e_{\boldsymbol{\alpha}}(\boldsymbol{x})\right)$ *равен* $2^{n-1} - 2^{\frac{n}{2}-1}$.

Доказательство. С учетом доказательства теоремы 1 достаточно доказать достижимость неравенства (7) для некоторой булевой функции $f(\boldsymbol{x})$. Положим

$$f(\boldsymbol{x}) = f\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right) = \psi\left(\boldsymbol{x}^{(1)}\right) \oplus \left\langle \boldsymbol{a}, \boldsymbol{x}^{(2)}\right\rangle,$$

$\boldsymbol{x}^{(1)} \in \mathbb{F}_2^{2m}$, $\boldsymbol{a}, \boldsymbol{x}^{(2)} \in \mathbb{F}_2^{2l}$, где $\psi\left(\boldsymbol{x}^{(1)}\right)$ — некоторая бент-функция. В соответствии с (3) имеем

$$\breve{f}(\boldsymbol{\alpha}) = \sum_{\boldsymbol{x}\in\mathbb{F}_2^n} (-1)^{f(\boldsymbol{x})}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})} =$$

$$= \sum_{\boldsymbol{x}^{(1)}\in\mathbb{F}_2^{2m}} (-1)^{\psi(\boldsymbol{x}^{(1)})}(-1)^{\langle\boldsymbol{\alpha}^{(1)},\boldsymbol{x}^{(1)}\rangle} \sum_{\boldsymbol{x}^{(2)}\in\mathbb{F}_2^{2l}} (-1)^{\varphi(\boldsymbol{x}^{(2)}\oplus\boldsymbol{\alpha}^{(2)})}(-1)^{\langle\boldsymbol{a},\boldsymbol{x}^{(2)}\rangle} =$$

$$= \left(\pm 2^m\right)\left(\pm 2^l\right) = \pm 2^{n/2}$$

для всех $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}) \in \mathbb{F}_2^n$, поскольку $\varphi\left(\boldsymbol{x}^{(2)}\right)$, $\psi\left(\boldsymbol{x}^{(1)}\right)$ — бент-функции. □

Таким образом, основные параметры произвольного кода Адамара совпадают с параметрами кода Рида — Маллера первого порядка. Перейдем теперь к сложности декодирования кода Адамара или, как уже отмечалось выше, сложности нахождения $\max\limits_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|\breve{f}(\boldsymbol{\alpha})\right|$. Для нахождения набора чисел $\breve{f}(\boldsymbol{\alpha})$, $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ необходимо в соответствии с (3) умножить матрицу $\left\|(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}\right\|_{\boldsymbol{\alpha},\boldsymbol{x}\in\mathbb{F}_2^n}$ на вектор $\left((-1)^{f(\boldsymbol{x})}\right)_{\boldsymbol{x}\in\mathbb{F}_2^n}$. В случае $e_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \langle\boldsymbol{\alpha},\boldsymbol{x}\rangle$ матрица $\left\|(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}\right\|_{\boldsymbol{\alpha},\boldsymbol{x}\in\mathbb{F}_2^n}$ хорошо факторизуется, что в свое время послужило основой для разработки алгоритма быстрого преобразования Фурье, который быстро умножал матрицу $\left\|(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}\right\|_{\boldsymbol{\alpha},\boldsymbol{x}\in\mathbb{F}_2^n}$ на вектор $\left((-1)^{f(\boldsymbol{x})}\right)_{\boldsymbol{x}\in\mathbb{F}_2^n}$. Рассмотрим другие случаи $e_{\boldsymbol{\alpha}}(\boldsymbol{x})$, когда возможна какая-либо факторизация матрицы $\left\|(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})}\right\|_{\boldsymbol{\alpha},\boldsymbol{x}\in\mathbb{F}_2^n}$.

Сравним теперь возможности аппроксимации булевых функций элементами двух биортогональных базисов: аффинными функциями $\langle\boldsymbol{\alpha},\boldsymbol{x}\rangle \oplus \varepsilon$ и функциями вида $\langle\boldsymbol{\alpha}^{(1)},\boldsymbol{x}^{(1)}\rangle \oplus \varphi\left(x^{(2)} \oplus \boldsymbol{\alpha}^{(2)}\right) \oplus \varepsilon$. Из предыдущего вытекает, что в общем случае для выбора лучшего варианта аппроксимации достаточно сравнить два числа $\max\limits_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|\widehat{f}(\boldsymbol{\alpha})\right|$ и $\max\limits_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|\breve{f}(\boldsymbol{\alpha})\right|$. Большее число укажет, элементами какого базиса данная функция приближается лучше. Реализация этого пути предполагает большой объем вычислений для нахождения $\max\limits_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|\widehat{f}(\boldsymbol{\alpha})\right|$ и $\max\limits_{\boldsymbol{\alpha}\in\mathbb{F}_2^n}\left|\breve{f}(\boldsymbol{\alpha})\right|$. Но, оказывается, что некоторые предварительные выводы можно сделать, анализируя одно полезное аналитическое соотношение между $\widehat{f}(\boldsymbol{\alpha})$ и $\breve{f}(\boldsymbol{\alpha})$.

Лемма 1. *Для любой булевой функции $f(\boldsymbol{x})$ и для любой системы функций $e_{\boldsymbol{\alpha}}(\boldsymbol{x}) = e_{(\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)})}\left(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)}\right) = \langle\boldsymbol{\alpha}^{(1)},\boldsymbol{x}^{(1)}\rangle \oplus \varphi\left(x^{(2)} \oplus \boldsymbol{\alpha}^{(2)}\right) \oplus \varepsilon$ из примера 1 выполнено соотношение*

$$\breve{f}(\boldsymbol{\alpha}) = \frac{1}{2^l} \sum_{\boldsymbol{\beta}^{(2)}\in\mathbb{F}_2^{2l}} (-1)^{\langle\boldsymbol{\beta}^{(2)},\boldsymbol{\alpha}^{(2)}\rangle}(-1)^{\varphi^*(\boldsymbol{\beta}^{(2)})}\widehat{f}\left(\boldsymbol{\alpha}^{(1)},\boldsymbol{\beta}^{(2)}\right), \tag{8}$$

где $\varphi^*(\boldsymbol{\beta}^{(2)})$ — дуальная бент-функция к функции $\varphi(\boldsymbol{x}^{(2)})$.

Доказательство. Подставим выражения для обратных преобразований Уолша — Адамара

$$(-1)^{f(\boldsymbol{x})} = \frac{1}{2^n} \sum_{\boldsymbol{\beta}\in\mathbb{F}_2^n} \widehat{f}(\boldsymbol{\beta})(-1)^{\langle\boldsymbol{\beta},\boldsymbol{x}\rangle},$$

$$(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})} = \frac{1}{2^n} \sum_{\boldsymbol{\gamma}\in\mathbb{F}_2^n} \widehat{e_{\boldsymbol{\alpha}}}(\boldsymbol{\gamma})(-1)^{\langle\boldsymbol{\gamma},\boldsymbol{x}\rangle}$$

в определение $\breve{f}(\boldsymbol{\alpha})$ (3) и выполним элементарные преобразования

$$\breve{f}(\boldsymbol{\alpha}) = \sum_{\boldsymbol{x}\in\mathbb{F}_2^n} (-1)^{f(\boldsymbol{x})}(-1)^{e_{\boldsymbol{\alpha}}(\boldsymbol{x})} =$$

$$= \sum_{\boldsymbol{x}\in\mathbb{F}_2^n} \left(\frac{1}{2^n}\sum_{\boldsymbol{\beta}\in\mathbb{F}_2^n} \widehat{f}(\boldsymbol{\beta})(-1)^{\langle\boldsymbol{\beta},\boldsymbol{x}\rangle}\right)\left(\frac{1}{2^n}\sum_{\boldsymbol{\gamma}\in\mathbb{F}_2^n} \widehat{e_{\boldsymbol{\alpha}}}(\boldsymbol{\gamma})(-1)^{\langle\boldsymbol{\gamma},\boldsymbol{x}\rangle}\right) =$$

$$= \frac{1}{2^{2n}} \sum_{\boldsymbol{\beta},\boldsymbol{\gamma}\in\mathbb{F}_2^n} \widehat{f}(\boldsymbol{\beta})\widehat{e_{\boldsymbol{\alpha}}}(\boldsymbol{\gamma}) \left(\sum_{\boldsymbol{x}\in\mathbb{F}_2^n} (-1)^{\langle\boldsymbol{\beta},\boldsymbol{x}\rangle}(-1)^{\langle\boldsymbol{\gamma},\boldsymbol{x}\rangle}\right) = \frac{1}{2^n}\sum_{\boldsymbol{\beta}\in\mathbb{F}_2^n} \widehat{f}(\boldsymbol{\beta})\widehat{e_{\boldsymbol{\alpha}}}(\boldsymbol{\beta}). \tag{9}$$

Значение $\widehat{e_{\boldsymbol{\alpha}}}(\boldsymbol{\beta})$ легко находится

$$\widehat{e_{\boldsymbol{\alpha}}}(\boldsymbol{\beta}) = \sum_{\substack{\boldsymbol{x}^{(1)}\in\mathbb{F}_2^k \\ \boldsymbol{x}^{(2)}\in\mathbb{F}_2^{2l}}} (-1)^{\langle\boldsymbol{\alpha}^{(1)},\boldsymbol{x}^{(1)}\rangle}(-1)^{\varphi(\boldsymbol{\alpha}^{(2)}\oplus\boldsymbol{x}^{(2)})}(-1)^{\langle\boldsymbol{\beta}^{(1)},\boldsymbol{x}^{(1)}\rangle}(-1)^{\langle\boldsymbol{\beta}^{(2)},\boldsymbol{x}^{(2)}\rangle} =$$

$$= \sum_{\boldsymbol{x}^{(1)}\in\mathbb{F}_2^k} (-1)^{\langle\boldsymbol{\alpha}^{(1)}\oplus\boldsymbol{\beta}^{(1)},\boldsymbol{x}^{(1)}\rangle} \sum_{\boldsymbol{x}^{(2)}\in\mathbb{F}_2^{2l}} (-1)^{\varphi(\boldsymbol{\alpha}^{(2)}\oplus\boldsymbol{x}^{(2)})}(-1)^{\langle\boldsymbol{\beta}^{(2)},\boldsymbol{x}^{(2)}\rangle} =$$

$$= \begin{cases} 0, & \text{если } \boldsymbol{\beta}^{(1)} \neq \boldsymbol{\alpha}^{(1)}, \\ (-1)^{\langle\boldsymbol{\beta}^{(2)},\boldsymbol{\alpha}^{(2)}\rangle}(-1)^{\varphi^*(\boldsymbol{\beta}^{(2)})}\cdot 2^{k+l}, & \text{если } \boldsymbol{\beta}^{(1)} = \boldsymbol{\alpha}^{(1)}. \end{cases}$$

Подставляя полученное выражение в (9), получим (8). □

Литература

[1] MacWilliams F. J., Sloane N. J. A. The Theory of Error-Correcting Codes. North-Holland Publishing Company, Amsterdam, New York, Oxford 1977. (Имеется русский перевод: Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов, исправляющих ошибки. «Связь», Москва, 1979).

Комбинирующие k-аффинные функции[5]

О. А. Логачёв, А. А. Сальников, В. В. Ященко

Булевы функции, используемые при построении комбинирующих генераторов (так называемые комбинирующие функции) должны обладать набором свойств, необходимых для того, чтобы соответствующий комбинирующий генератор был устойчив относительно ряда методов криптографического анализа (см. [2, 5, 7, 8]). Список криптографических свойств булевых функций постоянно пополняется, что обусловлено непрерывным развитием методов криптографического анализа. Ниже будет рассмотрено одно из таких свойств комбинирующих булевых функций.

Будем обозначать $\mathbb{F}_2$ — конечное поле из двух элементов, $\mathcal{F}_n$ — множество булевых функций от n переменных и $\overline{\mathcal{F}}_n$ — множество булевых функций из $\mathcal{F}_n$ существенно зависящих от всех n переменных (см. [4]), $\deg f$ — алгебраическая степень полинома Жегалкина функции f. Если $\deg f \leqslant 1$, то булева функция называется аффинной. Пусть $f \in \mathcal{F}_n$, для наборов $1 \leqslant i_1 < \ldots < i_k \leqslant n$, $\boldsymbol{b} = \left(b^{(1)}, \ldots, b^{(k)}\right) \in V_k$ обозначим $f^{b^{(1)},\ldots,b^{(k)}}_{i_1,\ldots,i_k}$ булеву функцию из $\mathcal{F}_{n-k}$, полученную из f фиксацией переменных $x^{(i_1)} = b^{(1)}, \ldots, x^{(i_k)} = b^{(k)}$.

Определение 1. Булева функция f из $\overline{\mathcal{F}}_n$ называется *k-аффинной*, $0 \leqslant k \leqslant n-1$, если существуют наборы $1 \leqslant i_1 < \ldots < i_k \leqslant n$, $\boldsymbol{b} = \left(b^{(1)}, \ldots, b^{(k)}\right) \in V_k$ такие, что $f^{b^{(1)},\ldots,b^{(k)}}_{i_1,\ldots,i_k}$ является аффинной: $\deg f^{b^{(1)},\ldots,b^{(k)}}_{i_1,\ldots,i_k} \leqslant 1$.

Любая аффинная функция является 0-аффинной, а любая функция из $\overline{\mathcal{F}}_n$ является $(n-1)$-аффинной.

Определение 2. Булева функция f из $\overline{\mathcal{F}}_n$ называется *сильно k-аффинной*, $0 \leqslant k \leqslant n-1$, если существует набор $1 \leqslant i_1 < \ldots < i_k \leqslant n$ такой, что для любого $\boldsymbol{b} = \left(b^{(1)}, \ldots, b^{(k)}\right) \in V_k$ $f^{b^{(1)},\ldots,b^{(k)}}_{i_1,\ldots,i_k}$ является аффинной: $\deg f^{b^{(1)},\ldots,b^{(k)}}_{i_1,\ldots,i_k} \leqslant 1$.

Ясно, что если f является сильно k-аффинной, то $ill(f) \leqslant k$ (см. [1]).

Определение 3. *Уровнем аффинности* $\mathrm{la}\, f$ булевой функции $f \in \mathcal{F}_n$ называется минимальное неотрицательное целое число k, для которого f является k-аффинной.

[5]Работа поддержана Российским фондом фундаментальных исследований (номера проектов 02-01-00581 и 02-01-00687).

Для любой функции f справедливо неравенство $\operatorname{la} f \leqslant n-1$; для аффинной функции f выполнено $\operatorname{la} f = 0$.

Предложение 1. *Пусть* $f \in \overline{\mathcal{F}}_n$, $\deg f \geqslant 2$, $l_{\boldsymbol{u},\varepsilon}(\boldsymbol{x}) = \bigoplus\limits_{i=1}^{n} u^{(i)}x^{(i)} \oplus \varepsilon$ — *аффинная функция из* $\mathcal{F}_n$. *Тогда*

$$\operatorname{la} f = \operatorname{la}(f \oplus l_{\boldsymbol{u},\varepsilon}) \ .$$

Доказательство. Пусть $\operatorname{la} f = k$. Тогда существуют наборы $1 \leqslant i_1 < \ldots < i_k \leqslant n$, $\boldsymbol{b} = \left(b^{(1)}, \ldots, b^{(k)}\right) \in V_k$ такие, что $f^{b^{(1)},\ldots,b^{(k)}}_{i_1,\ldots,i_k}$ является аффинной функцией. Следовательно функция $\Big(f \oplus \bigoplus\limits_{i=1}^{n} u^{(i)}x^{(i)} \oplus \varepsilon\Big)^{b^{(1)},\ldots,b^{(k)}}_{i_1,\ldots,i_k}$ также является аффинной и поэтому $\operatorname{la}\Big(f \oplus \bigoplus\limits_{i=1}^{n} u^{(i)}x^{(i)} \oplus \varepsilon\Big) \leqslant k = \operatorname{la} f$. Обратное неравенство доказывается аналогично. □

Предложение 2. *Пусть* f — k-*аффинная функция из* $\overline{\mathcal{F}}_n$. *Тогда для любой подстановки* $\mathfrak{g}$ *из группы Джевонса* $\mathfrak{D}_n$ *(см. [3]) булева функция* $f'(\boldsymbol{x}) = f^{\mathfrak{g}}(\boldsymbol{x}) = f(\mathfrak{g}\boldsymbol{x})$, $\boldsymbol{x} \in V_n$ *является* k-*аффинной.*

Пример 1. Рассмотрим квадратичную функцию

$$g\left(x^{(1)}, x^{(2)}, \ldots, x^{(n)}\right) = \bigoplus_{i<j} x^{(i)}x^{(j)} \oplus \bigoplus_{i=1}^{n} u^{(i)}x^{(i)} \oplus \varepsilon \ ,$$

где $\boldsymbol{u} = \left(u^{(1)}, \ldots, u^{(n)}\right) \in V_n$, $\varepsilon \in \mathbb{F}_2$. Ясно, что произвольная фиксация не более чем $n-2$ переменных приводит к функции, алгебраическая степень которой равна 2. Следовательно $\operatorname{la} g = n-1$, то есть в данном случае уровень аффинности принимает максимальное значение.

Предложение 3. *Пусть* $f \in \overline{\mathcal{F}}_n$, $g \in \overline{\mathcal{F}}_m$ — *функции от непересекающихся переменных. Тогда*

$$\operatorname{la}(f \oplus g) = \operatorname{la} f + \operatorname{la} g \ .$$

Для булевой функции $f \in \overline{\mathcal{F}}_n$ через $\Gamma(f)$ будем обозначать граф функции f. Это граф с n вершинами, помеченными числами $1, 2, \ldots, n$. Вершины с метками i и j соединены ребром в графе $\Gamma(f)$ тогда и только тогда, когда в полиноме Жегалкина функции f имеется моном, содержащий одновременно переменные $x^{(i)}$ и $x^{(j)}$. Кликой графа называют его максимальный полный подграф (см. [6]).

Предложение 4. *Пусть*

$$f\left(x^{(1)}, x^{(2)}, \ldots, x^{(n)}\right) = \bigoplus_{i<j} c^{(i,j)}x^{(i)}x^{(j)} \oplus \bigoplus_{i=1}^{n} u^{(i)}x^{(i)} \oplus \varepsilon \ ,$$

где $c^{(i,j)}, u^{(i)}, \varepsilon \in \mathbb{F}_2$. *Пусть граф* $\Gamma(f)$ *имеет клику мощности* d. *Тогда* $\operatorname{la} f > d-2$.

Булевы функции, используемые в качестве комбинирующей функции, в значительной степени определяют качества комбинирующего генератора. В частности, уровень аффинности $\operatorname{la} f$ комбинирующей функции f позволяет дать оценку трудоёмкости метода, предложенного в [2], определения ключей на основе рангового критерия.

Литература

[1] Логачёв О. А., Сальников А. А., Ященко В. В. Некоторые характеристики «нелинейности» групповых отображений, Дискретный анализ и исследование операций, Серия 1, **Том 8**, № 1, Январь–март, 2001, С. 40–54.

[2] Логачёв О. А., Сальников А. А., Ященко В. В. Корреляционная иммунность и реальная секретность, Тезисы выступления на конференции «Математика и безопасность информационных технологий», МГУ, Москва, 2003.

[3] Поваров Г. Н. О групповой инвариантности булевых функций. В сб.: Применение логики в науке и технике. Издательство АН СССР, Москва, 1960, С. 263–340.

[4] Яблонский С. В. Введение в теорию k-значной логики. Дискретная математика и математические вопросы кибернетики. **Том 1**, Москва, «Наука», 1974, С. 9–66.

[5] Chepyzhov V., Smeets B. On a Fast Correlation Attacks on Certain Stream Ciphers. Advances in Cryptology: EUROCRYPT'91 // Lect. Notes in Comput. Sci. — **Vol. 547.** — New York: Springer-Verlag. — 1991. — P. 176–185.

[6] Harary F. Graph Theory. Addison-Wesley publishing company, Reading, 1969.

[7] Meier W., Staffelbach O. Fast Correlation Attacks on Certain Stream Ciphers. Journal of Cryptology. — **Vol. 1.** — No. 3. — P. 159–176. — 1989.

[8] Siegentaler T. Decrypting a Class of Stream Ciphers Using Ciphertext Only. // IEEE Trans. on Computers. — 1985. — **Vol. C-34.** — 1. — P. 81–85.

Схемы открытого распределения ключа на основе некоммутативной операции. Использование в схемах данного типа символа степенного вычета

В. В. Назаров

Большинство известных на данный момент схем открытого распределения ключа опираются на сложность задачи дискретного логарифмирования. Самым известным примером является схема Диффи — Хеллмана (Diffie–Hellman).

В 1993 г. В. М. Сидельников [1] предложил новую, альтернативную, идею — использовать в схемах открытого распределения ключа некоммутативную операцию. Было приведено 2 варианта схемы:

Пусть есть полугруппа $(\mathbf{G}, *)$, где $*$ — некоммутативная, ассоциативная, полиномиально вычислимая операция. $\mathbf{G}_0, \mathbf{G}_1, \mathbf{G}_2$ — подполугруппы, внутри которых операция $*$ коммутирует. **A**, **B** — абоненты, $\gamma \notin \mathbf{G}_0$, K — общий секретный ключ.

Схема I

A	**B**
1. Выбирает $a_1, a_2 \in \mathbf{G}_0$, вычисляет $d_A = a_1 * \gamma * a_2$, отправляет d_A **B**.	1. Выбирает $b_1, b_2 \in \mathbf{G}_0$, вычисляет $d_B = b_1 * \gamma * b_2$, отправляет d_B **A**.
2. Вычисляет $K = K_A = a_1 * d_B * a_2$.	2. Вычисляет $K = K_B = b_1 * d_A * b_2$.

Схема II

A	**B**
1. Выбирает $a_i \in \mathbf{G}_i$,$i = 1, 2$; вычисляет $d_A = a_1 * a_2$; отправляет d_A **B**.	1. Выбирает $b_i \in \mathbf{G}_i$, $i = 1, 2$; вычисляет $d_B = b_1 * b_2$; отправляет d_B **A**.
2. Вычисляет $K = K_A = a_1 * d_B * a_2$.	2. Вычисляет $K = K_B = b_1 * d_A * b_2$.

Сразу стало ясно, что использование умножения матриц в качестве операции $*$ делает обе эти схемы нестойкими (из-за линейности). М. А. Черепнёв [2] предложил использовать операцию, основанную на символе Якоби в $\mathbb{Z}$. О. Н. Василенко заметил, что можно использовать не только его, но и обобщение символа Якоби (символ степенного вычета) в $\mathbb{Z}[\zeta_p]$.

В данном докладе будут рассмотрены некоторые свойства схемы при использовании в качестве $*$, операции, основанной на символе степенного вычета (он же — обобщенный символ Якоби).

Определение 1. $\zeta_p = e^{\frac{2\pi i}{p}}$, $\mathbb{Z}[\zeta_p]$ — кольцо целых чисел кругового поля, $\alpha \in \mathbb{Z}[\zeta_p]$, $\mathcal{J}$ — идеал в $\mathbb{Z}[\zeta_p]$, взаимнопростой с (p), тогда символ степенного вычета —

$$\left(\frac{\alpha}{\mathcal{J}}\right)_p = {\zeta_p}^i \equiv \alpha^{\frac{N(\mathcal{J})-1}{p}} \pmod{\mathcal{J}} \quad \text{для простого } \mathcal{J},$$

$$\left(\frac{\alpha}{\mathcal{J}}\right)_p = \prod_{\mathcal{J}_i | \mathcal{J};\ \mathcal{J}_i \text{ — простые}} \left(\frac{\alpha}{\mathcal{J}_i}\right)_p^{\nu_{\mathcal{J}_i}(\mathcal{J})}.$$

Тогда в качестве операции $*$ можно взять: $a * b = a \cdot b \cdot \left(\frac{\eta(a)}{\mu(b)}\right)_p$, где μ, η — мультипликативные функции, дающие на ζ_p 1 (см. [2]).

Мы покажем, что при данной операции $*$ схемы I и II принципиально различны, докажем теорему об эквивалентности разложения на множители остностельно $*$ и разложения в кольце $\mathbb{Z}[\zeta_p]$ и предложим алгоритм построения $\mathbf{G}_1, \mathbf{G}_2$ для произвольных функций μ, η

Теорема 1. *При данном выборе операции $*$ схема I является нестойкой.*

Утверждение теоремы следует из следующей формулы:

$$K = \frac{\frac{d_A}{\gamma} * d_B}{\frac{d_A}{\gamma} * \gamma} \cdot d_A.$$

Справедливость этой формулы проверяется выкладками, основанными на определении $*$, свойств функций μ, η и коммутирования элементов $\mathbf{G}_0$ (а точнее коммутирования элементов a_2 и b_1).

Теорема 2. *При данном выборе операции $*$ задача нахождения K в схеме II по открытой информации эквивалентна задаче нахождения по открытой информации любого из трех элементов:*

$$\tau_0 = \left(\frac{\eta(a_1)}{\mu(d_B)}\right)_p \cdot \left(\frac{\eta(d_B)}{\mu(a_2)}\right)_p; \quad \tau_1 = \frac{a_2 * b_1}{b_1 * a_2}; \quad \tau_2 = \frac{b_2 * a_1}{a_1 * b_2}.$$

В частности, если $\mu = \eta$, то τ_1 и τ_2 — символы норменного вычета.

Утверждение теоремы следует из формул:

$$K = d_A \cdot d_B \cdot \tau_0; \quad K = (d_A * d_B) \cdot {\tau_1}^{-1}; \quad K = (d_B * d_A) \cdot {\tau_2}^{-1}.$$

Таким образом в данной модели схема I является нестойкой, а схема II устойчива по отношению к аналогичной атаке, между тем ранее существенных отличий между схемами I и II известно не было.

Как указано в работе [2], если противник умеет решать задачу разложения на множители из $\mathbf{G}_1, \mathbf{G}_2$ относительно операции $*$, то он умеет находить ключ K в схеме II по открытой информации.

Теорема 3. *Пусть существует полиномиальный по x, y алгоритм нахождения символа степенного вычета $\left(\frac{x}{y}\right)_p$. Пусть G_1, G_2 таковы, что если $z \in G_i$, то $z \cdot {\zeta_p}^m \in G_i, m = 1, \ldots, p$. Тогда задача разложения на множители из G_1, G_2 относительно $*$ полиномиально эквивалентна задаче разложения на множители из G_1, G_2 в кольце $\mathbb{Z}[\zeta_p]$*

Утверждение теоремы следует из фактов, очевидных из определения $*$:

- если $r = r_1 \cdot r_2$, то $r = \left(r_1 \cdot \left(\frac{\eta(r_1)}{\mu(r_2)}\right)_p^{-1}\right) * r_2$;
- если $r = r_1 * r_2$, то $r = r_1 \cdot \left(r_2 \cdot \left(\frac{\eta(r_1)}{\mu(r_2)}\right)_p\right)$

Поскольку в общем случае задача разложения на множители относительно умножения в $\mathbb{Z}[\zeta_p]$ сложнее задачи разложения на множители в $\mathbb{Z}$, то выбирая подходящие подполугруппы $\mathbf{G}_1, \mathbf{G}_2$ можно надеяться на стойкость схемы относительно такого взлома (в силу сложности задачи разложения на простые в $\mathbb{Z}$ для некоторых случаев).

Одним из важных этапов в построении работающей схемы является построение коммутирующих подполугрупп $\mathbf{G}_1, \mathbf{G}_2$. Ниже будет предложен алгоритм их построения, основанный на следующем свойстве, замеченным М. А. Черепнёвым в статье [2].

Утверждение 1. *Если* $\frac{a*b}{b*a} = {\zeta_p}^m$; $\frac{a*c}{c*a} = {\zeta_p}^n$, *то* $\frac{a*(bc)}{(bc)*a} = {\zeta_p}^{m+n}$.

Алгоритм

1. Первый элемент e_1 выбирается случайным образом с соблюдением условий взаимной простоты $(\eta(e_1))$ и $(\mu(e_1))$ с (p).
2. Пусть уже построено множество $\mathbf{M} = \{e_1, \ldots, e_k\}$; $e_i * e_j = e_j * e_i$ для $i, j = 1, \ldots, k$; $i \neq j$.
   2.1. Выбираем $f_1, \ldots, f_{k+1}$ случайным образом из множества
   $$\mathbf{U} = \{x \mid ((\eta(x)), (\mu(e_i))) = ((\eta(e_i)), (\mu(x))) = ((\eta(x)), (p)) = ((\mu(x)), (p)) = 1,\ i = 1, \ldots, k\}.$$
   2.2. Считаем $s_{i,j} \in [0, p-1]$; $i = 1, \ldots, k$, $j = 1, \ldots, k+1$ такие, что $\frac{e_i * f_j}{f_j * e_i} = {\zeta_p}^{s_{i,j}}$; обозначим через $\mathbf{S}$ матрицу $\{s_{i,j}\}$.
   2.3. Если существует j_0: $s_{i,j_0} = 0$ $i = 1, \ldots, k$, то добавляем в $\mathbf{M}$ f_{j_0}.
   2.4. Иначе ищем нетривиальное решение системы $\mathbf{S} \cdot \overrightarrow{T} \equiv \overrightarrow{0} \pmod{p}$.
   2.5. Пусть $t_1, \ldots, t_{k+1}$ — нетривиальное решение, тогда добавляем в множество $\mathbf{M}$ элемент $e_{k+1} = {f_1}^{t_1} \cdot, \ldots, \cdot {f_{k+1}}^{t_{k+1}}$.

В силу утверждения 1 добавленный элемент коммутирует со всеми e_j. Нетривиальное решение существует, так как число неизвестных в системе больше числа уравнений. На построение h элементов необходимо $\mathcal{O}(h^4)$ операций по модулю p и $\mathcal{O}(h^3)$ вычислений символа степенного вычета и операций в $\mathbb{Z}[\zeta_p]$

Если мы построили $\mathbf{M} = \{e_1, \ldots, e_h\}$; то в качестве коммутативной подполугруппы можно использовать мультипликативные комбинации ${e_1}^{w_1}, \ldots, {e_h}^{w_h}$. с целыми показателями. Поскольку операция $*$ определена не для всех пар элеменов из построенных подполугрупп, то в схему нужно будет добавить проверку на определенность $*$. Заметим, что все образующие подполугруппы могут быть известны лишь общему вычислительному центру, а абоненты могут знать только выбранные случайным образом подмножества из множества образующих.

Данная модель представляется интересной как с точки зрения криптографии, так и с точки зрения возможности применения в ней разнообразных теоретико-числовых и вообще математических объектов. Из указанного выше следует, что использование схемы II предпочтительнее использования схемы I. Кроме того, описанные подходы к вскрытию схем приводят к задачам нахождения символа норменного вычета или задаче разложения на множители в $\mathbb{Z}[\zeta_p]$, решение которых в общем случае представляется достаточно сложным.

Литература

[1] Сидельников В. М., Черепнёв М. А., Ященко В. В. Системы открытого распределения ключей на основе некоммутативных полугрупп // Докл. акад. наук, 1993, 332, 5, с. 566–567.

[2] Черепнёв М. А. Схема открытого распределения ключа на основе некоммутативной группы // Дискр. мат., 2003.

Крипто-стеганографическая обработка данных на основе применения тригонометрических рядов с неубывающими коэффициентами

М. В. Корытова, Р. Т. Файзуллин

Рассматривается блочный алгоритм шифрования, основанный на применении тригонометрических рядов с неубывающими слагаемыми, с произвольно большой длиной блока при ограниченной длине ключа и предлагается дальнейшее использование этого алгоритма в стеганографии.

Алгоритм шифрования

Пусть имеется n символов сообщения. «Разместим» эти символы на единичной окружности, через шаги Δt_i, подчиненные некоторому закону. Например, в простейшем случае если мы проходим расстояние чуть большее чем 2π, по окружности, с равным шагом, то затем смещаемся по ходу движения (или увеличиваем шаг) на некоторый угол t_{shift}. Также, размещать символы можно не с нуля, а с некоторого заданного угла t_{begin}. Получим для каждого символа своё место на окружности, определённое некоторым углом t. Отображаем окружность в плоскость (X, Y), с помощью преобразования:

$$\begin{aligned} X &= \sum_{i=1}^{N} A_i \sin(AA_i t) + B_i \cos(BB_i t), \\ Y &= \sum_{i=1}^{N} C_i \sin(CC_i t) + D_i \cos(DD_i t). \end{aligned} \tag{1}$$

Здесь N, A_i, B_i, C_i, D_i, AA_i, BB_i, CC_i, DD_i $(i = 1, \ldots, N)$ — заданные числа, причем коэффициенты A_i, B_i, C_i D_i не убывают, а AA_i, BB_i, CC_i, DD_i наоборот, возрастают с ростом i. Далее мы проектируем точки полученной кривой, соответствующие точкам на окружности, с расположенными там символами, на заданную прямую $ax + by = d$, и таким образом уже на ней получаем перемешанный набор символов. Разработана программа, в которой реализован данный алгоритм шифрования. В качестве символов в программе берутся биты шифруемого файла. Исследование алгоритма с помощью критериев [1] и проверки на монотонность [1], на статистическую безопасность шифра, на устойчивость к ошибкам передачи и восстановления.

Стеганографический алгоритм

Пусть есть два изображения, причем одно больше другого. Можно рассмотреть точки меньшего из них как символы в криптоалгоритме и, с помощью преобразования (1), «разбросать» их по большей картине. При этом X и Y понимаются как координаты в большом изображении. Для восстановления скрытого изображения необходимо знать t_{begin}, t_{step}, t_{shift}, N, A_i, B_i, C_i, D_i, AA_i, BB_i, CC_i, DD_i для всех i, и размер скрытого рисунка в точках. Возможно и предварительное шифрование (перемешивание) скрываемой картинки опять же с помощью преобразования (1). Разработана программа, в которой реализован описанный алгоритм. Законный пользователь может не знать точное положение первой точки в контейнере. Для того чтобы избежать полного перебора, предлагается следующий стандартный стеганографический метод скрытия информации. В младшие биты скрываемого изображения записывается информация о черно-белом тестовом изображении. Таким образом, чтобы выяснить является ли точка первой, надо проверить, соответствуют ли данные в младших битых полученного изображения тому, что должно быть, при данной тестовой картине. Так в программе изменяется зелёная составляющая цвета точки, причём если в тестовом изображении точка чёрная то последний бит делается равным 0, а если не чёрная — 1.

Проверка на устойчивость

Проведена проверка к основным видам искажений, которым может подвергаться изображение в процессе хранения и передачи.

1. Преобразование файла к другому формату.

   1.1. BMP-JPEG-BMP. Алгоритм устойчив к данному преобразованию, но полученное скрытое сообщение может становиться чёрно-белым.

   1.2. BMP-GIF-BMP, BMP-LWF-BMP, BMP-PCX-BMP, BMP-PGM-BMP, BMP-PNG-BMP, BMP-PPM-BMP, BMP-TIF-BMP, BMP-TGA-BMP. Алгоритм устойчив к этим преобразованиям.

1.3. BMP-PBM-BMP. Алгоритм устойчив к данному преобразованию. Качество изображения портиться в соответствии с изменением стего.

1.4. BMP-EMF-BMP. Качество полученного изображения очень плохое, но при замене контейнера на однотонный качество резко улучшается.

2. Другие изменения.

2.1. Алгоритм устойчив к добавлению шума в стего, при этом шум появляется и в результате.

2.2. Устойчив к усилению резкости

2.3. Не устойчив к смазыванию.

2.4. Выдерживает сжатие до 30 процентов линейного размера контейнера, но размер стего перед поиском скрытого изображения должен быть восстановлен.

2.5. Выдерживает обрезание краёв, размер стего также должен быть восстановлен.

Важно отметить, что в одном контейнере может быть скрыта не одна картина.

Литература

[1] Кнут Д. Искусство программирования на ЭВМ. Т .2: Получисленные алгоритмы. М.: Мир, 1977, 483 с.

Экспоненциальные S-блоки

С. В. Агиевич, А. А. Афоненко

1 Введение

Пусть V_n — n-мерное векторное пространство над полем $\mathbb{F}_2 = \{0, 1\}$, везде далее $n > 1$. Базовыми компонентами многих симметричных криптосистем являются подстановки $\boldsymbol{s}\colon V_n \to V_n$, которые отвечают за сложную зависимость между прообразами, образами и ключами криптопреобразований. В контексте криптографических приложений такие подстановки принято называть S-блоками.

Основными критериями выбора S-блоков являются:

(a) малые значения разностных характеристик — по известному различию между прообразами $\boldsymbol{x}$ и $\boldsymbol{x}'$ трудно прогнозировать различие между образами $\boldsymbol{s}(\boldsymbol{x})$ и $\boldsymbol{s}(\boldsymbol{x}')$;

(b) высокая нелинейность — по известному значению линейной комбинации координат $\boldsymbol{x}$ трудно прогнозировать значение линейной комбинации координат $\boldsymbol{s}(\boldsymbol{x})$;

(c) высокие степени координатных булевых функций $\boldsymbol{s}$;

(d) распространение ошибок — изменение одной или нескольких координат $\boldsymbol{x}$ приводит к изменению каждой из координат $\boldsymbol{s}(\boldsymbol{x})$ с вероятностью близкой к $1/2$.

Нами предлагается способ построения подстановок $\boldsymbol{s}$, основанный на возведении в степень в поле $\mathbb{F}_{2^n}$ из 2^n элементов, и проводится анализ связанных с критериями (a)–(d) свойств таких *экспоненциальных* S-блоков.

2 Конструкция

Введем в рассмотрение функцию абсолютного следа $\mathrm{Tr}\colon \mathbb{F}_{2^n} \to \mathbb{F}_2$, $\beta \mapsto \beta + \beta^2 + \ldots + \beta^{2^{n-1}}$ и выберем некоторый базис $e_0, \ldots, e_{n-1}$ поля $\mathbb{F}_{2^n}$ над $\mathbb{F}_2$ (подробнее см. [1]). Элементу $x \in \mathbb{F}_{2^n}$ поставим в соответствие вектор $\boldsymbol{x} = (x_0, \ldots, x_{n-1}) \in V_n$ с координатами $x_i = \mathrm{Tr}(e_i x)$. В свою очередь, вектору $\boldsymbol{x}$ поставим в соответствие число $\overline{\boldsymbol{x}} = x_0 + 2x_1 + \ldots + 2^{n-1}x_{n-1}$ из множества $\{0, 1, \ldots, 2^n - 1\}$. Легко убедиться, что отображения $x \mapsto \boldsymbol{x}$ и $\boldsymbol{x} \mapsto \overline{\boldsymbol{x}}$ являются биективными.

Выберем примитивный элемент $\alpha \in \mathbb{F}_{2^n}$ с минимальным многочленом $f(x) \in \mathbb{F}_2[x]$, и рассмотрим отображение $s\colon V_n \to \mathbb{F}_{2^n}$, действующее по правилу

$$s(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} = \boldsymbol{0}, \\ \alpha^{\overline{\boldsymbol{x}}}, & \boldsymbol{x} \neq \boldsymbol{0}. \end{cases} \tag{1}$$

Так как $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2^n-1}$ суть все ненулевые элементы $\mathbb{F}_{2^n}$, то s является биекцией. Заменяя в (1) образы $s(\boldsymbol{x})$ векторами, получаем S-блок $\boldsymbol{s}\colon V_n \to V_n$, который будем называть экспоненциальным. Существует ровно

$$\frac{\varphi(2^n - 1)}{n}(2^n - 1)(2^n - 2)\ldots(2^n - 2^{n-1})$$

различных экспоненциальных S-блоков, действующих на V_n, отличающихся выбором примитивного многочлена $f(x)$ и базиса $e_0, \ldots, e_{n-1}$.

Подстановку $\boldsymbol{s}$ можно задать n координатными булевыми функциями $s_0(\boldsymbol{x}), \ldots, s_{n-1}(\boldsymbol{x})$ так, что $\boldsymbol{s}(\boldsymbol{x}) = (s_0(\boldsymbol{x}), \ldots, s_{n-1}(\boldsymbol{x}))$. Через $\mathcal{L}(\boldsymbol{s})$ обозначим линейную оболочку (с коэффициентами из поля $\mathbb{F}_2$) координатных функций $\boldsymbol{s}$.

3 Разностные характеристики

Пусть G_1, G_2 — конечные абелевы группы и s — биекция $G_1 \to G_2$. Введем в рассмотрение величину

$$\mathcal{R}(s) = \max_{\substack{a \in G_1, a \neq 0 \\ b \in G_2}} \sum_{x \in G_1} \mathbf{I}\{s(x + a) = s(x) + b\},$$

где $\mathbf{I}\{\mathcal{E}\}$ — индикатор наступления события $\mathcal{E}$. Очевидно, что $\mathcal{R}(s) \geqslant 2$.

Характеристика $\mathcal{R}(s)$ отражает эффективность методов разностного криптоанализа при использовании s в качестве функционального элемента криптосистемы. Малые значения $\mathcal{R}(s)$ затрудняют применение разностных методов.

Перенесем на V_n и обозначим через $\boxplus$ операцию сложения целых чисел по модулю 2^n: запись $\boldsymbol{c} = \boldsymbol{a} \boxplus \boldsymbol{b}$ для $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in V_n$ означает, что $\overline{\boldsymbol{c}} = \big(\overline{\boldsymbol{a}} + \overline{\boldsymbol{b}}\big) \bmod 2^n$. Операции $\oplus$ (обычное сложение в V_n и $\mathbb{F}_{2^n}$) и $\boxplus$ часто используются при построении криптосистем. Поэтому интерес представляет исследование характеристики $\mathcal{R}(s)$ при выборе в качестве G_1 и G_2 групп $\langle V_n, \oplus\rangle$, $\langle \mathbb{F}_{2^n}, \oplus\rangle$, $\langle V_n, \boxplus\rangle$. Если, например, $G_1 = \langle V_n, \boxplus\rangle$, $G_2 = \langle \mathbb{F}_{2^n}, \oplus\rangle$, то вместо $\mathcal{R}(s)$ пишем $\mathcal{R}_{\boxplus\oplus}(s)$.

В литературе встречаются конструкции биективных отображений s, для которых $\mathcal{R}_{\oplus\oplus}(s) = 2$ или $\mathcal{R}_{\boxplus\boxplus}(s) = 2$ (см. соответственно [3] и [2]). Предлагаемая нами конструкция (1) является близкой к оптимальной в смысле минимума характеристики $\mathcal{R}_{\boxplus\oplus}(s)$, о чем свидетельствует следующий результат.

Теорема 1. *Если $f(x)$ не делит ни один из многочленов вида*

$$x^{2^{n-1}} + x^t + 1, \quad t = 1, \ldots, 2^{n-1} - 1,$$

то $\mathcal{R}_{\boxplus\oplus}(s) \leqslant 3$ для отображения (1). *В противном случае $\mathcal{R}_{\boxplus\oplus}(s) = 4$.*

При переходе от отображения s вида (1) к S-блоку $\boldsymbol{s}$ используется базис $e_0, \ldots, e_{n-1}$. Легко проверить, что при любом выборе базиса $\mathcal{R}_{\boxplus\oplus}(\boldsymbol{s}) = \mathcal{R}_{\boxplus\oplus}(s)$ и $\mathcal{R}_{\oplus\oplus}(\boldsymbol{s}) = \mathcal{R}_{\oplus\oplus}(s)$.

4 Нелинейность

Обозначим через L_n множество аффинных булевых функций от n переменных, т. е. функций вида $l(\boldsymbol{x}) = \boldsymbol{b} \cdot \boldsymbol{x} + c = b_0 x_0 + b_1 x_1 + \ldots + b_{n-1} x_{n-1} + c$, $\boldsymbol{b} \in V_n$, $c \in \mathbb{F}_2$. Пусть множество L_n^* получено из L_n удалением нулевой функции. Напомним [3], что нелинейностью $\boldsymbol{s}$ называется величина

$$\mathcal{N}(\boldsymbol{s}) = \operatorname{dist}(L_n^*, \mathcal{L}(\boldsymbol{s})) = \min_{\substack{l(\boldsymbol{x}) \in L_n^* \\ \sigma(\boldsymbol{x}) \in \mathcal{L}(\boldsymbol{s})}} \operatorname{dist}(l(\boldsymbol{x}), \sigma(\boldsymbol{x})),$$

где $\operatorname{dist}(l(\boldsymbol{x}), \sigma(\boldsymbol{x})) = \sum_{\boldsymbol{x} \in V_n} \mathbf{I}\{l(\boldsymbol{x}) \neq \sigma(\boldsymbol{x})\}$ есть расстояние Хэмминга между таблицами истинности функций $l(\boldsymbol{x})$ и $\sigma(\boldsymbol{x})$. При использовании $\boldsymbol{s}$ в составе криптосистемы большие значения нелинейности повышают стойкость к методам линейного анализа.

Следующая теорема дает оценку нелинейности снизу.

Теорема 2. *Пусть* $r = 2^n - 1$, $K(\boldsymbol{b})$ — *множество индексов ненулевых координат вектора* $\boldsymbol{b} \in V_n$ *и*

$$\Pi(\boldsymbol{b}) = \frac{1}{r} \sum_{h=1}^{r-1} \prod_{k \in K(\boldsymbol{b})} \left| \operatorname{tg} \frac{\pi 2^k h}{r} \right|.$$

Для экспоненциального S-блока $\boldsymbol{s}\colon V_n \to V_n$ *справедлива оценка*

$$\mathcal{N}(\boldsymbol{s}) \geqslant 2^{n-1} - 1 - 2^{n/2-1} \max_{\substack{\boldsymbol{b} \in V_n \\ \boldsymbol{b} \neq \boldsymbol{0}}} \Pi(\boldsymbol{b}).$$

Нам не удалось найти приемлемых оценок сверху для суммы $\Pi(\boldsymbol{b})$. Прямые вычисления показывают, что $\Pi(\boldsymbol{b})$ существенно зависит от веса Хэмминга $\operatorname{wt}(\boldsymbol{b})$ вектора $\boldsymbol{b}$, т. е. от числа ненулевых координат $\boldsymbol{b}$. Как правило, $\Pi(\boldsymbol{b})$ максимальна, если $\operatorname{wt}(\boldsymbol{b}) = n$.

5 Степени координатных функций

Ненулевая функция $\sigma(\boldsymbol{x}) = \sigma(x_0, \ldots, x_{n-1}) \in \mathcal{L}(\boldsymbol{s})$ задается многочленом из кольца $\mathbb{F}_2[x_0, \ldots, x_{n-1}]$. При использовании $\boldsymbol{s}$ в составе криптосистемы желательно, чтобы степень $\deg(\sigma)$ данного многочлена была велика.

Теорема 3. *Если* $\boldsymbol{s}\colon V_n \to V_n$ — *экспоненциальный S-блок, то для любой ненулевой функции* $\sigma(x_0, \ldots, x_{n-1}) \in \mathcal{L}(\boldsymbol{s})$ *справедливо*

$$\deg(\sigma) \geqslant n - \lceil \log_2(n+1) \rceil$$

где $\lceil z \rceil$ *есть минимальное целое* $\geqslant z$.

Следующая теорема дает критерий выбора α, при котором степени всех ненулевых координатных функций $\boldsymbol{s}$ достигают максимального значения $n-1$. Напомним, что $a \in \mathbb{F}_{2^n}$ — нормальный элемент над $\mathbb{F}_2$, если набор $a, a^2, \ldots, a^{2^{n-1}}$ является базисом $\mathbb{F}_{2^n}$ над $\mathbb{F}_2$.

Теорема 4. *Если* $\boldsymbol{s}\colon V_n \to V_n$ — *экспоненциальный S-блок, то* $\deg(\sigma) = n-1$ *для всех ненулевых функций* $\sigma(x_0, \ldots, x_{n-1}) \in \mathcal{L}(\boldsymbol{s})$ *тогда и только тогда, когда* $a = \alpha(1+\alpha)^{-1}$ *является нормальным элементом над* $\mathbb{F}_2$.

6 Распространение единичных ошибок

Пусть $\sigma(\boldsymbol{x}) \in \mathcal{L}(\boldsymbol{s})$. Оценим вероятность изменения значения $\sigma(\boldsymbol{x})$ при изменении j-й координаты $\boldsymbol{x}$, т. е. вероятность

$$p_j(\sigma) = \mathsf{P}\{\sigma(x_0, \ldots, x_j, \ldots, x_{n-1}) \neq \sigma(x_0, \ldots, x_{j-1}, x_j + 1, x_{j+1}, \ldots, x_{n-1})\},$$

которая вычисляется в предположении, что $\boldsymbol{x}$ есть случайный вектор с равномерным на V_n распределением.

Теорема 5. *Если* $\boldsymbol{s}\colon V_n \to V_n$ *— экспоненциальный* S*-блок, то для любой ненулевой функции* $\sigma(\boldsymbol{x}) \in \mathcal{L}(\boldsymbol{s})$ *и всех* $j = 0, 1, \ldots, n-1$ *справедливо неравенство*

$$\left| p_j(\sigma) - \frac{1}{2} \right| < \frac{\ln r}{\pi 2^{n/2}} + \frac{1}{2^{n/2+1}} + \frac{1}{2^{n-1}}, \quad r = 2^n - 1.$$

7 Заключение

Значение $s(\boldsymbol{x})$ можно вычислить, затратив не более $2n$ умножений в поле $\mathbb{F}_{2^n}$. Однако при наличии дополнительной памяти количество умножений можно сократить.

Действительно, пусть $m \mid n$ и $n = dm$. Вычислим и сохраним значения

$$T_i(t) = \alpha^{t2^{mi}}, \quad i = 0, \ldots, d-1, \quad t = 0, \ldots, 2^m - 1.$$

Теперь для $\overline{\boldsymbol{x}} = 2^{(d-1)m} t_{d-1} + \ldots + 2^m t_1 + t_0 \neq 0$, $0 \leqslant t_i < 2^m$, справедливо

$$s(\boldsymbol{x}) = \prod_{i=0}^{d-1} T_i(t_i)$$

и требуется выполнить только $d-1$ умножение. Для хранения таблиц $T_i(t)$ требуется $2^m n^2/m$ битов памяти. Например, при выборе $n = 32$, $m = 8$ потребуется 4096 байтов, что вполне приемлемо при программной реализации криптографических алгоритмов. При этом значения $s(\boldsymbol{x})$ вычисляются всего за 3 умножения в поле $\mathbb{F}_{2^{32}}$, а размерность $n = 32$ является достаточно большой для S-блоков, используемых в современных симметричных криптосистемах.

Литература

[1] Лидл Р., Нидеррайтер Г. Конечные поля. Т 1, 2. М.: Мир, 1988.

[2] Massey J. L. SAFER K-64: A byte-oriented block ciphering algorithm. Fast Software Encryption — Cambridge Security Workshop Proceedings, Lecture Notes in Computer Science, v. 809, 1994, p. 1–16.

[3] Nyberg K. Differential uniform mappings for cryptography. Advances in Cryptology: Proceedings of Eurocrypt 93, Lecture Notes in Computer Science, v. 765, 1993, p. 55–64.

О мономиальных базисах[6]

А. Ю. Серебряков

Известно, что тождества Мак-Вильямс имеют большое значение в теории кодирования [1]. В. М. Сидельников предложил обобщение тождеств Мак-Вильямс для некоторого класса орбитных групповых кодов [2], и представляется важным распространить эти результаты на более широкий класс групп.

Пусть G — конечная группа, $\rho : G \to GL(V)$ — ее конечномерное унитарное представление. Обозначим $g\boldsymbol{v} = \rho(g)\boldsymbol{v}$ для $g \in G$, $\boldsymbol{v} \in V$. Определим пространство X как орбиту некоторого вектора $\boldsymbol{a} \in V$: $X = G\boldsymbol{a}$. Пусть H — подгруппа группы G. Тогда орбитный код $\mathcal{C}$ в пространстве X — это орбита вектора $\boldsymbol{a}$ под действием группы H, т.е. $\mathcal{C} = H\boldsymbol{a}$.

Проблема нахождения тождества типа Мак-Вильямс для орбитного кода $\mathcal{C}$ сводится [3, 4] к проблеме поиска «мономиального» базиса в пространстве многочленов на конечном множестве $X_0 = G_0 v_0$ ($v_0 \in V_0$), а именно такого базиса (над $\mathbb{C}$) $\varphi_1, \ldots, \varphi_r \in \mathbb{C}[X_0]$, что выполнено:

1) для любого $g \in G$ выполнено $g\varphi_i = \chi_i(g)\varphi_{j(g,i)}$, $\chi_i(g) \in \mathbb{C}$;

[6]Работа выполнена при поддержке гранта РФФИ № 02-01-00687, и гранта INTAS № 00-738.

2) элементы этого базиса образуют группу по умножению, т. е.

$$\varphi_i \cdot \varphi_j = \varphi_{k(i,j)}.$$

Известно [2], что таким путем получаются тождества Мак-Вильямс для абелевых групп и для неприводимого двухмерного представления группы кватернионов Q_8.

Далее будем рассматривать вариант мономиального базиса, определенного непосредственно на группе G. Этот базис определим как набор функций $f_i : G \to \mathbb{C}$, $i = 1, \ldots, |G|$, удовлетворяющий следующим свойствам:

1) элементы f_i — линейно независимы (над $\mathbb{C}$);

2) элементы этого базиса образуют группу по умножению, т. е.

$$f_i \cdot f_j = f_{k(i,j)};$$

3) для любого $g \in G$ выполнено $gf_i = \chi_i(g) f_{j(g,i)}$, $\chi_i(g) \in \mathbb{C}$;

4) все функции, задающие одномерные представления группы G, являются элементами данного базиса;

5) $f_i(e) = 1$ для всех i, где e — единица группы G;

6) элементы мономиального базиса попарно ортогональны относительно скалярного произведения $(F_1, F_2) = \frac{1}{|G|} \sum_{g \in G} F_1(g)\overline{F_2(g)}$.

Очевидно, что для случая когда группа G абелева, мономиальный базис существует и однозначно определен — его элементами являются все характеры группы G

В общем случае элементы мономиального базиса удобно записывать как линейные комбинации матричных элементов неприводимых представлений группы G.

Довольно очевидно, что если для существует мономиальный базис для группы G, то он существует и для любой орбиты $\mathcal{C} = Ga$ этой группы, имеющей тривиальный стабилизатор: положим $\varphi_i(ga) = f_i(g)$.

Если существует мономиальный базис $f_1(x), \ldots, f_r(x)$ для группы G_0, то произведения вида $f_{i_1}(x_1) \cdot \ldots \cdot f_{i_n}(x_n)$ образуют мономиальный базис для группы G, являющейся прямым произведением n экземпляров группы G_0:

$$G = G_0 \times \ldots \times G_0 = \{(x_1, \ldots, x_n) : x_i \in G_0,\ i = 1, \ldots, n\}.$$

Элементы мономиального базиса удобно индексировать элементами абелевой группы A, которую образует данный мономиальный базис относительно умножения, так чтобы было выполнено $f_\alpha \cdot f_\beta = f_{\alpha \oplus \beta}$ ($\alpha, \beta \in A$, $\alpha \oplus \beta$ — сумма элементов α и β в группе A).

Нами разобраны случаи, когда группа $G = A_4$ (знакопеременная группа порядка 12), Q_8 (группа кватернионов порядка 8), D_n (группа диэдра порядка $2n$). Для каждой из перечисленных групп доказано, что мономиальный базис существует, и его элементы могут быть явно выписаны как линейные комбинации матричных элементов неприводимых представлений. Существование мономиального базиса позволяет выписывать для рассмотренных матричных групп соотношения типа Мак-Вильямс.

Абелеву группу, которую образуют элементы мономиального базиса по умножению, можно рассматривать как своего рода двойственный объект к исходной группе G_0, ее порядок совпадает с $|G_0|$. Для левого регулярного представления групп A_4, Q_8, D_{2m}, D_{2m+1} этим путем получены соответственно абелевы группы $\mathbb{Z}_3 \oplus \mathbb{Z}_4$, $\mathbb{Z}_2 \oplus \mathbb{Z}_4$, $\mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_m$, $\mathbb{Z}_2 \oplus \mathbb{Z}_{2m+1}$, которые мы будем рассматривать как в некотором смысле «двойственные» к исходным.

В пространстве конечномерного представления конечной мономиальной группы над полем $\mathbb{C}$ существует базис $v_1, \ldots, v_r$ такой, что для любого $g \in G$ и любого $i \in \{1, \ldots, r\}$ существует $j \in \{1, \ldots, r\}$, для которого выполнено $gv_i = \chi_i(g) v_j$. Это следует из того, что любое неприводимое представление мономиальной группы G является индуцированным с некоторого одномерного представления подгруппы $H \subset G$ [3]. Остается открытым вопрос, всегда ли можно выбрать базис в пространстве $\mathbb{C}[X_0]$

функций на орбите X_0 заданной конечной мономиальной группы G_0 таким, чтобы его элементы образовывали еще и группу по умножению (элементы умножаются как функции). Можно предположить, что таким путем получаются тождества типа Мак-Вильямс для любой мономиальной группы. В частности, это так, если группа G_0 — полупрямое произведение двух абелевых групп.

Далее будем рассматривать случай, когда $G = G_0 \times \ldots \times G_0$ (n раз) — прямое произведение, $\rho_0 : G_0 \to GL(V_0)$ — (конечномерное) представление группы G_0, а ρ — представление группы G в пространстве $V = V_0 \oplus \ldots \oplus V_0$, $\rho(g_1, \ldots, g_n)(v_1, \ldots, v_n) = (\rho_0(g_1)v_1, \ldots, \rho_0(g_n)v_n)$ ($g_i \in G_0$, $v_i \in V_0$). Для простоты будем считать, что стабилизатор вектора $a_0 \in V_0$ при действии группы G_0 тривиален.

Выпишем тождества типа Мак-Вильямс, когда для группы G_0 существует мономиальный базис f_α, $\alpha \in A$ ($f_{\alpha_1} \cdot f_{\alpha_2} = f_{\alpha_1 \oplus \alpha_2}$), а начальный вектор $a = (a_0, \ldots, a_0)$.

Будем рассматривать орбитные коды, которые получаются с помощью подгрупп, задаваемых некоторым набором $\Sigma = \{\sigma_1(x), \ldots, \sigma_u(x)\}$ элементов мономиального базиса $\sigma_s(x) = \prod_{j=1}^n f_{\alpha_j^s}(x_j)$ ($s = 1, \ldots, u$). Множество Σ задает подгруппу $H = H_\Sigma \subset G$ с помощью условия $h \in H$, тогда и только тогда, когда $\sigma_s(hx) = \sigma_s(x)$ для любого $x \in X$, $s = 1, \ldots, u$.

Для орбитного кода $\mathcal{C} = Ha$ определим его двойственный код $\mathcal{C}^*$ как подмножество (подгруппу) абелевой группы A^n элементов, представимых в виде (целочисленных) линейных комбинаций

$$\mathcal{C}^* = \{r_1 k_1 + \ldots + r_u k_u\} \subset A^n,$$

где векторы $k_s = (\alpha_1^s, \ldots, \alpha_n^s) \in A^n$, коэффициенты $r_s \in \mathbb{Z}$, $s = 1, \ldots, u$.

Для данного орбитного кода $\mathcal{C} = Ha$ определим функцию, порожденную его всеми попарными расстояниями:

$$\Phi_{\mathcal{C}}(z) = \frac{1}{|\mathcal{C}|} \sum_{c,b \in \mathcal{C}} \exp(|b - c|^2 z) = \sum_{b \in \mathcal{C}} \exp((2 - 2\,\mathrm{Re}(a, b))z)$$

($\mathrm{Re}\, u$ — вещественная часть комплексного числа u, (a, b) — G-инвариантное скалярное произведение в пространстве V).

Данную функцию можно считать обобщением весового энумератора. Далее вместо нее будем изучать тета-функцию

$$\Theta_{\mathcal{C}}(z) = \sum_{b \in \mathcal{C}} \exp(\mathrm{Re}((a, b)z)$$

которая легко выражается через функцию $\Phi(z) = e^2 \Theta_{\mathcal{C}}(-2z)$.

Теорема (тождество МакВильямс для функции $\Theta_{\mathcal{C}}(z)$)**.** *Если существует мономиальный базис для группы G_0, то*

$$\Theta_{\mathcal{C}}(z) = \frac{1}{|\mathcal{C}^*|} \sum_{j_1, \ldots, j_l} B_{j_1, \ldots, j_l} \varphi_1(z)^{j_1} \cdot \ldots \cdot \varphi_l(z)^{j_l},$$

где l — число различных функций в множестве $\left\{\widehat{f}_\alpha : \alpha \in A\right\}$;

$\varphi_i(z) = \sum_{x \in X_0} f_{\gamma_i}(x) \exp(\mathrm{Re}(x, a_0)z) = \widehat{f}_{\gamma_i}(z)$, $i = 1, \ldots, l$;
$B_{j_1, \ldots, j_l} = |\{\overline{\alpha} \in \mathcal{C}^* : \nu_s(\overline{\alpha}) = j_s,\ s = 1, \ldots, l\}|$;
$\nu_s(\overline{\alpha}) = \left|\left\{m \in \{1, \ldots, n\} : \widehat{f}_{\alpha_m} = \varphi_s\right\}\right|$, $s = 1, \ldots, l$, $\overline{\alpha} = (\alpha_1, \ldots, \alpha_n) \in A^n$.

Доказательство. Имеем $\sigma_s(x) = \prod_{j=1}^n f_{\alpha_j^s}(x_j)$, $s = 1, \ldots, u$. Обозначим

$$P_s(x) = 1 + \sigma_s(x) + \ldots + \sigma_s^{d-1}(x),$$

где d — наименьшее общее кратное порядков элементов группы A.

Тогда можно записать характеристическую функцию кода $\mathcal{C}$ как

$$\chi_{\mathcal{C}}(x) = \frac{1}{d^u} \prod_{s=1}^u P_s(x) = \frac{1}{d^u} \sum_{\overline{\gamma} \in \{0,1,\ldots,d-1\}^u} \sigma_1^{\gamma_1}(x) \cdot \ldots \cdot \sigma_u^{\gamma_u}(x)$$
$$= \frac{1}{d^u} \sum_{\overline{\alpha} = r_1 k_1 + \ldots + r_u k_u} x^{\overline{\alpha}},$$

где $x^{\overline{\alpha}} = \prod_{j=1}^{n} f_{\alpha_j}$. В последней сумме каждое слагаемое имеет кратность r, где $r = r(\Sigma) = d^u/|\mathcal{C}^*|$ — число различных представлений вектора $0 \in A^n$ в виде суммы векторов k_s, $s = 1, \ldots, u$, с коэффициентами из $\{0, 1, \ldots, d-1\}$, откуда

$$\chi_{\mathcal{C}}(x) = \frac{1}{|\mathcal{C}^*|} \sum_{\overline{\alpha} \in \mathcal{C}^*} x^{\overline{\alpha}}.$$

Имеем

$$\Theta_{\mathcal{C}}(z) = \sum_{x \in X} \chi_{\mathcal{C}}(x) e^{\mathrm{Re}(a,x)z} = \frac{1}{|\mathcal{C}^*|} \sum_{x \in X} \sum_{\overline{\alpha} \in \mathcal{C}^*} x^{\overline{\alpha}} e^{\mathrm{Re}(a,x)z},$$

и, меняя порядок суммирования, получаем

$$\Theta_{\mathcal{C}}(z) = \frac{1}{|\mathcal{C}^*|} \sum_{\overline{\alpha} \in \mathcal{C}^*} \prod_{j=1}^{n} \sum_{x_j \in X_0} f_{\alpha_j}(x_j) e^{\mathrm{Re}(a_0, x_j)z} = \frac{1}{|\mathcal{C}^*|} \sum_{\overline{\alpha} \in \mathcal{C}^*} \prod_{j=1}^{n} \varphi_{\alpha_j}(z),$$

откуда, очевидно, вытекает утверждение теоремы. □

Литература

[1] Мак-Вильямс Ф. Дж., Слоэн Н. Дж. Теория кодов, исправляющих ошибки. М.: Связь, 1979.

[2] Sidel'nikov V. M. MacWilliams-type identities for linear p-ary codes in non-Hamming spaces. Seventh International Workshop on Algebraic and Combinatorial Theory, 18–24 June 2000, Bansko, Bulgaria, p. 275–278.

[3] Кириллов А. А. Элементы теории представлений. М.: Наука, 1972.

[4] Serebryakov A., Sidelnikov V. On MacWilliams-tipe identities for orbit codes. Proc. Eighth International Workshop on Algebraic and Combinatorial Coding Theory, 8–14 September 2002, Tsarskoe Selo, Russia, p. 232–233.

О групповых свойствах криптоалгоритма Веста

М. А. Пудовкина

В данной работе рассмотрены групповые свойства алгоритма поточного шифрования Веста, зависящего от параметров, конкретизируя которых получаем, в частности, криптоалгоритмы Веста-2, Веста-2М. Криптоалгоритмы Веста-2, Веста-2М предложены фирмой ЛАН-Крипто [1]. Их анализу посвящены, например, работы [2, 3, 4].

Пусть $p, d, n, l, t_1, t_2 \in \mathbb{N}$, где $\mathbb{N}$ - множество натуральных чисел и $0 \leqslant t_1 < t_2 \leqslant d-1$.

Криптоалгоритм Веста моделируются автономным автоматом $A_{\mathrm{B}} = (\mathbb{Z}_p^d \times \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}, \mathbb{Z}_{2^l}, F, f)$. Функция переходов $F\colon \mathbb{Z}_p^d \times \mathbb{Z}_{2^n} \to \mathbb{Z}_p^d \times \mathbb{Z}_{2^n}$ определяется двумя преобразованиями $\mu\colon \mathbb{Z}_p^d \times \mathbb{Z}_{2^n} \to \mathbb{Z}_{2^n}$, $\nu\colon \mathbb{Z}_p^d \times \mathbb{Z}_{2^n} \to \mathbb{Z}_{2^n}$ и будет описана ниже.

Состоянием автомата A_{B} такте $t \geqslant 1$ является тройка $((x_{t+d-}1, \ldots, x_t), w_t, v_t) \in \mathbb{Z}_p^d \times \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n}$, где $(x_{t+d-1}, \ldots, x_t)$ - d-мерный вектор над кольцом $\mathbb{Z}_p$, являющейся состоянием линейного регистра сдвига в такте $t \geqslant 1$ с функцией обратной связи $g(x) = x^d - a_{d-1}x^{d-1} - \ldots - a_0 \pmod p$, $a_j \in \mathbb{Z}_p$, $j = \overline{0, p-1}$. Начальное состояние $((x_{d-1}, \ldots, x_0), w_0, v_0)$ автомата A_{B} является ключом криптоалгоритма Веста.

Приведем описание i-го ($i = 1, 2 \ldots$) такта работы A_{B}.

Функция переходов F:

$$\begin{aligned} x_{i+d} &= a_{d-1}x_{i+d-1} + \ldots + a_0 x_i \pmod p, \\ w_i &= \mu(x_{i+t_2}, w_{i-1}), \\ v_i &= \nu(x_{i+t_1}, v_{i-1}). \end{aligned}$$

Пусть $\mu_i(w) = \mu(i, w)$, $\nu_i(w) = \nu(i, w)$ для любых пар $(i, w) \in \mathbb{Z}_p \times \mathbb{Z}_{2^n}$, $\nu\colon i \to i+1 \pmod{2^n}$, $S(X)$ - симметрическая группа подстановок, действующая на множестве X, $S_n = S(\mathbb{Z}_n)$.

Пусть биективное отображение $\psi\colon \mathbb{Z}_2^n \to \mathbb{Z}_{2^n}$ определяется равенством:

$$\psi\colon (a_{n-1}, \ldots, a_0) \to \sum_{i=0}^{n-1} 2^i a_i.$$

Пусть S_n действует на координатах n-мерных двоичных векторов естественным образом. Обозначим через $\widetilde{\pi}$ подстановку степени 2^n, соответствующую подстановке $\pi \in S_n$ и переводящую каждый элемент $a = (a_{n-1}, \ldots, a_0) \in \mathbb{Z}_2^n$ в элемент $b = (b_{n-1}, \ldots, b_0) \in \mathbb{Z}_2^n$, где $b_{j\pi} = a_j$. Таким образом, группа S_n индуцирует подгруппу $\Omega(\mathbb{Z}_2^n)$, изоморфную S_n, группы $S(\mathbb{Z}_2^n)$, действующую на n-мерных двоичных векторах.

Пусть группа $\Omega(\mathbb{Z}_{2^n}) \leqslant S(\mathbb{Z}_{2^n})$ подобна группе$\Omega(\mathbb{Z}_2^n)$, где подобие задается парой отображений (ψ, ρ), $\rho\colon \widetilde{\pi} \to \widehat{\pi}$, $\widehat{\pi} = \psi^{-1}\widetilde{\pi}\psi$.

Криптоалгоритмы Веста-2, Веста-2М, во введенной модели, имеют вид:

Криптоалгоритм Веста-2М

$n = 16$, p - простое число порядка 2^{15}, $d = 31$, $g(x) = x^{31} + x^{10} - 1$, $t_1 = 0$, $t_2 = 10$, $\nu_j = \eta^j$, $\mu_j = \widehat{\pi}\eta^j$, где $j = \overline{0, p-1}$, $\pi \in S_n$, причем $\pi\colon i \to i+1 \pmod{n}$.

Криптоалгоритм Веста-2

$n = 16$, p - простое число порядка 2^{15}, $d = 31$, $g(x) = x^{31} - x^3 - 1$, $t_1 = 0$, $t_2 = 3$, $\nu_j = \eta^j$, $\mu_j = eta^j$, $j = \overline{0, p-1}$.

Нам понадобится для описания свойств группы криптоалгоритма Веста следующее утверждение.

Утверждение 1. *Пусть $m = 2^n$, $n \in \mathbb{N}$, $\eta \in S_m$, причем $\eta\colon i \to i+1 \pmod{m}$, и π - полноцикловая подстановка из S_n. Тогда подгруппа $\langle \widehat{\pi}, \eta \rangle$ из $\Omega(\mathbb{Z}_{2^n})$ примитивна.*

Доказательство. Предположим, что группа $\langle \widehat{\pi}, \eta \rangle$ импримитивна. Из критерия примитивности (см., например, [5]) следует, что все системы блоков импримитивности циклической группы $\langle \eta \rangle$ являются орбитами ее подгрупп. Поэтому для любого $\alpha \in \mathbb{Z}_m$ и $r = 0 \pmod 2$ орбита $\alpha\langle \eta^r \rangle$, являющаяся циклом подстановки η^r, есть блок импримитивности циклической группы $\langle \eta \rangle$.

Если группа $\langle \widehat{\pi}, \eta \rangle$ импримитивна, то существует такое $r' = 0 \pmod 2$, что циклы подстановки $\eta^{r'}$ есть блоки импримитивности группы $\langle \widehat{\pi}, \eta \rangle$. Легко увидеть, что циклы подстановки η^r, где $r = 0 \pmod 2$, содержат только четные или нечетные элементы.

Пусть $\Delta_0 = 0\langle \eta^{r'} \rangle$, $\Delta_{m-1} = (m-1)\langle \eta^{r'} \rangle$, $1 < |\Delta_0| < m$. Поскольку $\widehat{\pi}\colon 0 \to 0$, $\widehat{\pi}\colon m-1 \to m-1$, то $\Delta_0^{\widehat{\pi}} = \Delta_0$, $\Delta_{m-1}^{\widehat{\pi}} = \Delta_{m-1}$ и $\Delta_{m-1} \cap \Delta_0 = \varnothing$. Пусть $x \in \Delta_0 \setminus \{0\}$. Тогда для $k = \overline{0, n-1}$ имеем: $x\widehat{\pi}^k \in \Delta_0$ и вес $|x\widehat{\pi}^k| = |x|$. Однако Δ_0 принадлежат только четные элементы, а среди n элементов, полученных полноцикловой перестановкой координат x, встречаются как четные так и нечетные элементы. Поэтому Δ_0 содержит также нечетные элементы, противоречие. Утверждение доказано. □

Теорема 1. *Пусть $m = 2^n$, n - четное число и группа $\langle \widehat{\pi}, \eta \rangle$ из $S(\mathbb{Z}_{2^n})$ примитивна. Тогда для произвольных фиксированных чисел $j_1, j_2 \in \mathbb{N}$ таких, что их разность $j_1 - j_2$ нечетна, группы $\langle \widehat{\pi}, \eta \rangle$, $\langle \widehat{\pi}\eta^{j_1}, \widehat{\pi}\eta^{j_2} \rangle$ и S_m изоморфны.*

Доказательство. Покажем сначала, что $\langle \widehat{\pi}, \eta \rangle \cong \langle \widehat{\pi}\eta^{j_1}, \widehat{\pi}\eta^{j_2} \rangle$. Поскольку $(\widehat{\pi}\eta^{j_2})^{-1} = \eta^{-j_2}\widehat{\pi}^{-1}$, то $\eta^{-j_2}\widehat{\pi}^{-1}\widehat{\pi}\eta^{j_1} = \eta^{j_1 - j_2}$. Так как $|j_1 - j_2| = 1 \pmod 2$, то $\langle \eta^{j_1 - j_2} \rangle = \langle \eta \rangle$. Отсюда $\widehat{\pi}\eta^{j_2}\eta^{-j_2} = \widehat{\pi}$. Поэтому $\widehat{\pi}, \eta \in \langle \widehat{\pi}\eta^{j_1}, \widehat{\pi}\eta^{j_2} \rangle$. Следовательно, группы $\langle \widehat{\pi}, \eta \rangle$ и $\langle \widehat{\pi}\eta^{j_1}, \widehat{\pi}\eta^{j_2} \rangle$ изоморфны.

В [6] доказано, что примитивная группа подстановок $G \subset S_m$, содержащая полный цикл, или совпадает с S_m, или изоморфна проективной линейной группе $\mathrm{PSL}(2, p)$, где $p = m - 1$ есть простое число. Поскольку в нашем случае число $m-1$ составное, то $\langle \widehat{\pi}, \eta \rangle$ изоморфна S_m. Теорема доказана. □

Для преобразования, используемого в криптоалгоритме Веста-2, опишем порождаемую им группу.

Следствие 1. *Пусть $m = 2^n$, n - четное число, $\eta \in S_m$, причем $\eta\colon i \to i+1 \pmod{m}$, а $\pi \in S_n$, причем $\pi\colon i \to i+1 \pmod{n}$. Тогда группы $\langle \widehat{\pi}, \eta \rangle$ и S_m изоморфны.*

Доказательство следует из утверждения 1 и теоремы 1.

В теореме 2 описана группа криптоалгоритма Веста.

Теорема 2. *Пусть* $p, n, d \in \mathbb{N}$, $\lambda(g,p,d)$ *- подстановка степени* p^d, *реализуемая автономным регистром сдвига длины* d *с функцией обратной связи* g *над кольцом* $\mathbb{Z}_p$. *Пусть* $\mu_j, \nu_j \in S_{2^n}$, $j = \overline{0,p-1}$, *и* $\delta_{t_1,t_2} \colon \mathbb{Z}_p^d \to S_{2^n} \times S_{2^n}$, *где* $(t_1,t_2) \in \mathbb{Z}_d \times \mathbb{Z}_d$, $t_1 < t_2$, $(a,b)^{\vec{x}\delta_{t_1,t_2}} = (a\mu_{x_{t_1}}, b\nu_{x_{t_2}})$ *и для любого* $\vec{x} \in \mathbb{Z}_p^d$. *Пусть* $M = \langle \mu_j \mid j = \overline{0,p-1} \rangle$, $\Gamma = \langle \nu_j \mid j = \overline{0,p-1} \rangle$. *Тогда группа криптоалгоритма Веста есть сплетение прямого произведения групп подстановок* $M \times \Gamma$ *и циклической группы* $\lambda(g,p,n)$, *причем*

$$((a,b)\vec{x})^{(\delta_{t_1,t_2},\lambda(g,p,n))} = \left((a,b)^{\vec{x}\delta_{t_1,t_2}}, \vec{x}\lambda(g,p,n)\right).$$

Следствие 2. *Пусть* $n \in \mathbb{N}$, p *- простое число,* λ_p *- подстановка степени* p^31, *реализуемая автономным регистром сдвига длины* 31 *с функцией обратной связи* $g(x) = x^{31} + x^{10} - 1$ *над кольцом* $\mathbb{Z}_p$. *Пусть* $\eta \colon i \to i+1 \pmod{2^n}$, $\nu_j = \eta^j$, $\mu_j = \widehat{\pi}\eta^j$, *где* $j = \overline{0,p-1}$, $\pi \in S_n$, *причем* $\pi \colon i \to i+1 \pmod{n}$, $\delta_{0,10} \colon \mathbb{Z}_p^{31} \to S_{2^{16}} \times S_{2^{16}}$, $(a,b)^{\vec{x}\delta_{0,10}} = (a\mu_{x_0}, b\nu_{x_{10}})$ *для любого* $\vec{x} \in \mathbb{Z}_p^{31}$. *Тогда группа криптоалгоритма Веста-2M есть сплетение прямого произведения групп* $S_{2^{16}} \times \mathbb{Z}_{2^{16}}$ *и циклической группы* $\langle \lambda_p \rangle$, *изоморфной* $\mathbb{Z}_{p^{31}-1}$, *причем*

$$((a,b)\vec{x})^{(\delta_{0,10},\lambda_p)} = \left((a,b)^{\vec{x}\delta_{0,10}}, \vec{x}\lambda_p\right).$$

Следствие 3. *Пусть* $n \in \mathbb{N}$, p *- простое число,* λ_p *- подстановка степени* p^31, *реализуемая автономным регистром сдвига длины* 31 *с функцией обратной связи* $g(x) = x^{31} - x^3 - 1$ *над кольцом* $\mathbb{Z}_p$. *Пусть* $\eta \colon i \to i+1 \pmod{2^n}$, $\nu_j = \eta^j$, $\mu_j = \eta^j$, $j = \overline{0,p-1}$, $\delta_{0,3} \colon \mathbb{Z}_p^{31} \to S_{2^{16}} \times S_{2^{16}}$, $(a,b)^{\vec{x}\delta_{0,3}} = (a\mu_{x_0}, b\nu_{x_3})$ *для любого* $\vec{x} \in \mathbb{Z}_p^{31}$. *Тогда группа криптоалгоритма Веста-2 есть сплетение прямого произведения групп* $\mathbb{Z}_{2^{16}} \times \mathbb{Z}_{2^{16}}$ *и циклической группы* $\langle \lambda_p \rangle$, *изоморфной* $\mathbb{Z}_{p^{31}-1}$, *причем*

$$((a,b)\vec{x})^{(\delta_{0,3},\lambda_p)} = \left((a,b)^{\vec{x}\delta_{0,3}}, \vec{x}\lambda_p\right).$$

Литература

[1] ОСТ 51-06-98. Алгоритм кодирования данных.

[2] Варфоломеев А. А., Жуков А. Е., Пудовкина М. А. Поточные криптосистемы. Основные свойства и методы анализа стойкости, М.: МИФИ, 2000.

[3] Pudovkina M. Cryptanalysis of the Vesta-2M Stream Cipher. Eurocrypt'01 (rump session), May 2001, `http://eprint.iacr.org/`, report 2001/043.

[4] Пудовкина М. А. О слабых состояниях криптосистемы ВЕСТА-2. В сб. тезисов конференции «Методы и технические средства обеспечения безопасности информации», Санкт-Петербург, 2002.

[5] Погорелов Б. А. Основы теории групп подстановок. I. Общие вопросы. Москва, 1986.

[6] Погорелов Б. А. Примитивные группы подстановок, содержащие 2^m-цикл. Н.: Наука, Алгебра и логика, т. 19, № 2, 1980.

Опыт криптографии сельских жителей, женщин и детей

С. С. Титов, Л. Г. Чукалова

Под шифрами «сельских жителей, женщин и детей» (по терминологии де ла Порта [1]) будем понимать упрощенные варианты реально применяемых систем шифрования и кодирования. Их использование оправдано как в учебном процессе, так и при анализе криптостойкости стандартных систем.

Вполне в согласии с тезисами [2] «такие криптосистемы как шифры Цезаря, Плейфера, Хилла, Вернама и другие полезны для изучения только с методической точки зрения» и «история криптографии крайне увлекательна и достойна отдельного рассмотрения», этот материал и материалы различных сайтов, в том числе `Cryptography.ru`, а также популярных и даже художественных книг, широко используется в процессе обучения студентов специальностей «Информационные технологии» и «Защита информации» Уральского государственного университета путей сообщения уже с первого курса.

Читаемые на младших курсах дисциплины «Дискретная математика», «Математическая логика» (с элементами теории алгоритмов и аппаратной реализации булевых функций) включают в себя элементы криптографии, криптоанализа, стеганографии, теории конечных автоматов и являются также подготовительными к курсу «Информационная безопасность» [3].

Рассмотрение вопросов нормативно-правовой базы информационной безопасности, освоение практических навыков организации защищенного документооборота предприятия проводится с помощью активных методов обучения в виде деловой игры, которая вовлекает студентов в коллективный труд с персональной ответственностью. В такой искусственно созданной среде изучаются также методы дешифрования исторических шифров [1] и усеченных аналогов современных криптосистем.

Вместе с тем эта форма позволяет подвести студентов к получению научных и научно-практических результатов [4, 5].

Так, проведено исследование поточных шифров $RC4(n)$ для малых n. Эти результаты используются, в том числе, при изучении (студентами специальности «ИТ») атак на систему защиты Windows 95/98 (утилита glide, см., например, [6]).

Получено удобное описание БЧХ-кодов, программно реализованы операции алгебры многочленов над конечными полями характеристики два [7], позволяющие эффективно изучать при помощи «криптографических многочленов» [8] линейные рекуррентные последовательности [1, 9], алгоритмы гаммирования с блоками усложнения [10], Rijndael и др. Предложены различные варианты процедур аутентификации с нулевым разглашением. Использование «детских» версий криптосистем, например RSA с малыми простыми числами, развивает интуицию обучающихся (в том числе теоретико-числовую [11]) и убеждает в реальности соответствующих атак.

Литература

[1] Бабаш А. В., Шанкин Г. П. Криптография. Под редакцией И. П. Шерстюка, Э. А. Применко / Серия книг «Аспекты защиты». М.: СОЛОН-Р. 2002. 512 с.

[2] Баричев С. Г., Гончаров В. В., Серов Р. Е. Основы современной криптографии. Учебный курс. М.: Горячая линия — Телеком. 2001. 120 с.

[3] Яковлев В. В., Корниенко А. А. Информационная безопасность и защита информации в корпоративных сетях железнодорожного транспорта. Учебник для вузов ж.-д. транспорта / Под ред. В. В. Яковлева. М.: УМК МПС России. 2002. 328 с.

[4] Титов С. С., Баданова О. М., Иициксон М. А., Торгашова А. В. Логарифм Зеха-Якоби в задаче расшифровки. Проблемы теоретической и прикладной математики. Труды 33-й Региональной молодежной конференции. Екатеринбург, 2002. С. 51–55.

[5] Титов С. С., Усольцев А. В., Кузнецова И. С., Демкина О. Е. Разностные схемы в моделировании декодера. Проблемы теоретической и прикладной математики. Труды 33-й Региональной молодежной конференции. Екатеринбург, 2002. С. 56–60.

[6] Крысин А. В. Информационная безопасность. Практическое руководство. М.: СПАРРК, Киев: ВЕК+. 2003. 320 с. + CD.

[7] Ленг С. Алгебра. М.: Мир. 1968. 564 с.

[8] Шнайер Б. Прикладная криптография. 2-е издание: протоколы, алгоритмы и исходные тексты на языке С. 1996. (пер. с англ.: Bruce Schneier. Applied Cryptography / Second Edition: Protocols, Algorithms, and Source Codes in C. John Wiley & Sons. 1996. 758 p.)

[9] Лидл Р., Нидеррайтер Г. Конечные поля. М.: Мир. 1988.

[10] Масленников М. Е. Практическая криптография. СПб: БХВ-Петербург. 2003. 464 с. + CD.

[11] Бухштаб А. А. Теория чисел. М.: Просвещение. 1966. 384 с.

Стандартные базисы полиномиальных идеалов над коммутативным артиновым цепным кольцом и их приложения[7]

Е. В. Горбатов, Д. А. Михайлов, А. В. Михалёв, А. А. Нечаев

1 Введение

Пусть $R[X] = R[x_1, \ldots, x_k]$ — коммутативное кольцо многочленов над кольцом R от переменных $x_1, \ldots, x_k$. Решение большого количества прикладных задач, связанных с кольцом $R[X]$, сводится к задаче проверки принадлежности заданного многочлена $F(X) \in R[X]$ некоторому идеалу $I \lhd R[X]$, заданному какой-нибудь системой образующих ψ. Истоки развития теории в направлении решения этой задачи лежат в работах Л. Кронекера, который нашел алгоритм построения стандартного базиса полиномиального идеала в кольце $\mathbb{Z}[x_1, \ldots, x_k]$ (см. [10]), представив, тем самым, эффективное доказательство теоремы Гильберта о базисе.

Важный промежуточный этап — результаты А. И. Ширшова и Д. Бухбергера, решающие задачу в случае, когда $R = P$ — поле. В этом случае на множестве $[X] = \{\boldsymbol{x}^{\boldsymbol{a}} : \boldsymbol{a} \in \mathbb{N}_0^k\}$ всех мономов задается некоторое полное упорядочение $\preccurlyeq$, согласованное с операцией умножения (*допустимое упорядочение*). Это упорядочение однозначно продолжается до частичного упорядочения $\preccurlyeq$ на множестве $[P, X]$ одночленов из $P[X]$, которое, в свою очередь, позволяет ввести понятие *старшего члена* многочлена и определить алгоритм *редуцирования (деления с остатком)* многочлена $F \in P[X]$ с помощью многочлена $G \in P[X]$ или конечной системы многочленов $\psi \subset P[X]$. С использованием этого алгоритма, и сопутствующего ему аппарата S-полиномов, по системе образующих ψ идеала $I \lhd P[X]$ строится система полиномов $\chi \subseteq I$, обладающая тем свойством, что любой полином $F \in P[X]$ редуцируется с помощью этой системы к единственному остатку, который равен 0, если $F \in I$. Такая система называется *базисом Ширшова — Грёбнера* или *стандартным базисом* идеала I (при данном алгоритме деления с остатком).

Для произвольного коммутативного нетерова кольца R известно существование стандартных базисов идеалов из $R[X]$, см., например, [14], где схема построения базиса формально повторяет изложенную выше схему для полей. В частности, продолжение порядка $\preccurlyeq$ на множестве мономов до частичного порядка на множестве $TR[X]$ одночленов $\alpha\boldsymbol{x}^{\boldsymbol{a}}$, $\alpha \in R$, никак не учитывало специфики коэффициентов α. Этот же подход использовался позже в работах [20, 18].

Однако, в случае, когда R не является полем, его элементы «неоднородны», в связи с наличием собственных идеалов, и иногда указанную неоднородность удается «учесть», задав на R нетривиальную нормирующую функцию и определяя продолжение порядка $\preccurlyeq$ на множество $TR[X]$, согласуясь с этой функцией. Это приводит к построению других стандартных базисов полиномиальных идеалов, которые мы будем называть *согласованными (с заданной нормирующей функцией)*. Последние оказываются весьма эффективными при решении ряда прикладных задач в теории кодирования и криптографии.

[7]Исследования были частично поддержаны Грантами Президента РФ НШ-2358.2003.9, НШ-1910.2003.1 и Грантами РФФИ 99-01-00382, 99-01-00941.

2 Предварительные замечания

Пусть R — кольцо из заглавия с радикалом Джекобсона (нильрадикалом) $\mathfrak{N} = \mathrm{Rad}(R)$. Тогда, если $\mathfrak{N} \neq 0$, то $\mathfrak{N} = \pi R$ для любого $\pi \in \mathfrak{N} \setminus \mathfrak{N}^2$ и для некоторого $n \in \mathbb{N}$ решетка всех идеалов R есть цепь:

$$R > \mathfrak{N} > \mathfrak{N}^2 > \ldots > \mathfrak{N}^{n-1} > \mathfrak{N}^n = 0, \quad gN^s = \pi^s R, \quad s \in \overline{0,n}. \tag{1}$$

Определим, следуя [2], *нормы* элемента $r \in R$, полинома $F \in R[X]$ и подмножества $\chi \subset R[X]$ равенствами:

$$\begin{aligned} \|r\| &= \max\left\{ i \in \overline{0,n} \;\middle|\; r \in \pi^i R \right\}; \\ \|F\| &= \max\left\{ i \in \overline{0,n} \;\middle|\; F \in \pi^i R[X] \right\}; \\ \|\chi\| &= \max\left\{ i \in \overline{0,n} \;\middle|\; \chi \subseteq \pi^i R[X] \right\}. \end{aligned} \tag{2}$$

Согласно общему подходу к построению стандартных базисов, надо последовательно определить: допустимый порядок на мономах, порядок на одночленах, старший член полинома, и, наконец, алгоритм редуцирования полинома полиномом.

Напомним следующее

Определение 1. Пусть $(U, \cdot)$ — полугруппа и $\preccurlyeq$ — порядок на U. Тройка $(U, \cdot, \preccurlyeq)$ называется *упорядоченной полугруппой* если

$$\forall\, a, b, c \in U : (a \preccurlyeq b) \implies (ac \preccurlyeq bc) \;\&\; (ca \preccurlyeq cb).$$

Если $\preccurlyeq$ — линейный (полный) порядок на U, то говорят, что $(U, \cdot, \preccurlyeq)$ — *линейно (вполне) упорядоченная полугруппа.*

Пусть $[X] = [x_1, \ldots, x_k] = \{\boldsymbol{x}^{\boldsymbol{a}} : \boldsymbol{a} \in \mathbb{N}_0^k\}$ — полугруппа коммутативных мономов над X. Очевидно, что $([X], \cdot, |)$ — упорядоченная полугруппа (здесь $|$ обозначает отношение делимости), изоморфная упорядоченной полугруппе $(\mathbb{N}_0^k, +, \leqslant)$, где $+$ означает покомпонентное сложение, а порядок $\leqslant$ получается из обычного порядка на $\mathbb{N}_0$ по формуле

$$(a_1, \ldots, a_k) \leqslant (b_1, \ldots, b_k) \overset{\text{def}}{\iff} a_1 \leqslant b_1, \ldots, a_k \leqslant b_k. \tag{3}$$

Предложение 1 (см., например, [15], с. 99). *Линейно упорядоченная полугруппа* $([X], \cdot, \preccurlyeq)$ *является вполне упорядоченной полугруппой если и только если* $\forall\, u \in [X] : 1 \preccurlyeq u$. *В этом случае порядок* $\preccurlyeq$ *называется* допустимым.

Пусть $F \in R[X]$ и $u \in [X]$. Элемент из R, являющийся коэффициентом при мономе u в полиноме F, обозначим через $\mathrm{Cf}(F, u)$. *Носителем* многочлена $F \in R[x]$ назовем множество

$$\mathrm{supp}(F) = \{u \in [X] \mid \mathrm{Cf}(F, u) \neq 0\}. \tag{4}$$

При заданном допустимом порядке $\preccurlyeq$ на $[X]$, любой ненулевой полином $F \in R[X]$ имеет непустой носитель в котором содержится наибольший относительно $\preccurlyeq$ моном u_0. Более того, полином F может быть представлен в виде $F = \alpha_0 u_0 + \ldots + \alpha_m u_m$, где $\alpha_1, \ldots, \alpha_m$ — ненулевые элементы из R и $u_1, \ldots, u_m$ — мономы, такие что $u_1 \succ \ldots \succ u_m$[8]). Исходя из этого представления, можно определить:

$$\begin{aligned} \mathrm{Hm}(F) &= u_1 \;\text{ — } \textit{старший моном } F; \\ \mathrm{Hc}(F) &= \alpha_1 \;\text{ — } \textit{старший коэффициент } F; \\ \mathrm{Ht}(F) &= \alpha_1 u_1 \;\text{ — } \textit{старший член } F. \end{aligned} \tag{5}$$

Именно эти функции использовались в большинстве предыдущих работ при построении редукций на полиномах и стандартных базисов идеалов (см. [10, 14, 18, 20]). В отличие от этого подхода,

[8]Полагаем, что вместе с порядком $\preccurlyeq$ заданы и порядки $\succcurlyeq$, $\prec$, $\succ$, при этом, например, $a \succ b \iff (b \preccurlyeq a) \;\&\; (a \neq b)$.

в [2], а затем и в [3, 4, 5, 6], было, по существу, использовано другое упорядочение на одночленах — упорядочение, *согласованное с нормой* (2) кольца R. А именно, для любых $\alpha u, \beta v \in [R, X]$:

$$\alpha u \preccurlyeq_{\mathrm{norm}} \beta v \overset{\mathrm{def}}{\iff} \begin{cases} \|\alpha\| > \|\beta\| \text{ или} \\ \|\alpha\| = \|\beta\| \text{ и } u \preccurlyeq v. \end{cases} \tag{6}$$

Отметим, что порядок $\preccurlyeq_{\mathrm{norm}}$ в общем случае не будет линейным. Например в кольце $\mathbb{Z}_4[x]$ одночлены x и $3x$ несравнимы (при единственном допустимом порядке $\preccurlyeq$ на $[x]$). Тем не менее, ограничение порядка $\preccurlyeq_{\mathrm{norm}}$ на *полугруппу* π*-мономов* $[\pi, X] = [\pi, x_1, \ldots, x_k] = \{\pi^a u \mid a \in \overline{0,n}; u \in [X]\}$, превращает $[\pi, X]$ во вполне упорядоченную полугруппу. Более того, имеет место следующее

Предложение 2 ([6], 2.4). *Пусть на* $[\pi, X]$ *задан порядок* $\leqslant$ *такой, что* $([\pi, X], \cdot, \leqslant)$ *— вполне упорядоченная полугруппа и* $0 \preccurlyeq U$ *для любого* $U \in [\pi, X]$. *Пусть также* $\preccurlyeq$ *— допустимый порядок, являющийся ограничением* $\leqslant$ *на* $[X]$. *Тогда* $\leqslant$ *совпадает с* $\preccurlyeq_{\mathrm{norm}}$.

Любой ненулевой полином $F \in R[X]$ может быть записан в виде $F = \beta_1 v_1 + \ldots + \beta_m v_m$, где $\beta_1, \ldots, \beta_m$ — ненулевые элементы из R, $v_1, \ldots, v_m$ — мономы, причем $\beta_1 v_1 \succcurlyeq_{\mathrm{norm}} \ldots \succcurlyeq_{\mathrm{norm}} \beta_m v_m$. Исходя из этого представления, в работах [3, 4, 5, 6] были введены следующие (согласованные с нормой (2) кольца R) функции:

$$\begin{aligned} \mathrm{Lm}(F) &= v_1 \text{ — } \textit{ведущий моном } F; \\ \mathrm{Lc}(F) &= \beta_1 \text{ — } \textit{ведущий коэффициент } F; \\ \mathrm{Lt}(F) &= \beta_1 v_1 \text{ — } \textit{ведущий член } F. \end{aligned} \tag{7}$$

Для произвольного подмножества $\chi \subseteq R[X]$ будем полагать $\mathrm{Lm}(\chi) = \{\mathrm{Lm}(G) \mid G \in \chi\}$. Аналогичные соглашения принимаем и для остальных введенных выше функций.

Эти функции обладают важными свойствами мультипликативности, которые отсутствуют у функций (5):

Предложение 3 ([6], 2.9). *Пусть даны два полинома* $F, G \in R[X]$, *такие, что* $FG \neq 0$, *тогда выполняются соотношения*

$$\begin{aligned} \mathrm{Lm}(FG) &= \mathrm{Lm}(F)\,\mathrm{Lm}(G), \\ \mathrm{Lc}(FG) &= \mathrm{Lc}(F)\,\mathrm{Lc}(G), \\ \mathrm{Lt}(FG) &= \mathrm{Lt}(F)\,\mathrm{Lt}(G). \end{aligned}$$

Прежде чем приступить к описанию конкретных алгоритмов редуцирования и стандартных базисов, дадим некоторые общие определения.

Определение 2. Полином $F \in R[X]$ называется *нормальным* относительно системы полиномов χ (при данном алгоритме редуцирования), если его невозможно редуцировать с помощью χ или если F не меняется при всяком редуцировании.

Любой полином $F \in R[X]$ может быть редуцирован относительно системы χ к некоторому полиному H, нормальному относительно χ. Совокупность всех таких полиномов H называется *множеством нормальных форм* полинома F относительно системы χ и обозначается $\mathrm{Nor}_\chi(F)$.

3 Согласованные стандартные базисы

В указанных выше работах, при построении стандартных базисов, использовалось понятие ведущего члена полинома (7) и следующее определение редукций.

Определение 3. Будем говорить, что полином $F \in R[X] = R[x_1, \ldots, x_k]$ *редуцируется* к полиному $H \in R[X]$ с помощью полинома $G \in R[X]$ и монома $u \in [X]$, и писать $F \xrightarrow[(G,u)]{} H$, если $\mathrm{Cf}(F, u\,\mathrm{Lm}(G)) = b\,\mathrm{Lc}(G)$ и $H = F - buG$.

Отметим, что данное определение не зависит от выбора элемента b. Действительно, для другого элемента $b' \in R$ такого, что $\mathrm{Cf}(F, u\,\mathrm{Lm}(G)) = b'\,\mathrm{Lc}(G)$ будем иметь:

$$(b - b')\,\mathrm{Lc}(G) = 0 \implies (b - b')G = 0 \implies F - buG = F - b'uG$$

.

Приведем, в удобной для нашего изложения форме, следующие определения.

Определение 4. Подмножество $\mathcal{F} \subseteq [X]$ называется *диаграммой Ферре* если для любых $u, v \in [X]$

$$(u \in \mathcal{F}) \ \&\ (v \mid u) \implies v \in \mathcal{F}.$$

Обозначим через $\overline{\alpha}$, $\overline{F}$ и $\overline{\psi}$ образы элемента $\alpha \in R$, полинома $F \in R[X]$ и системы полиномов $\psi \subseteq R[X]$ при естественных эпиморфизме $R \to \overline{R} = R/\,\mathrm{Rad}(R)$ и индуцированном им эпиморфизме $R[X] \to \overline{R}[X]$.

Определение 5 ([3], п. 2)**.** Пусть даны система полиномов $\psi \subseteq R[X]$ и диаграмма Ферре $\mathcal{F}$. Система ψ называется *круллевой системой с опорной диаграммой Ферре* $\mathcal{F}$, если

1) $\overline{\psi}$ — редуцированный базис Грёбнера-Ширшова идеала $(\overline{\psi})$ над полем $\overline{R}$,

2) $|\psi|$ и $|\overline{\psi}|$ имеют одинаковую мощность,

3) $[X] \setminus \mathcal{F} = \mathrm{Lm}(\psi)[X]$ — идеал в полугруппе мономов $[X]$ порожденный $\mathrm{Lm}(\psi)$,

4) $\mathrm{supp}(F - \mathrm{Lt}(F)) \subseteq \mathcal{F}$, для любого $F \in \psi$.

Следующая теорема была доказана в [2] при $k = 1$, в [3] для случая унитарного идеала $I \lhd R[X]$ и в [4, 5] для произвольного полиномиального идеала.

Теорема 1 ([2, 3, 4, 5])**.** *Пусть $I \lhd R[X]$ — произвольный идеал. Тогда для некоторого $t \in \overline{0, n-1}$ существуют цепочка*

$$\mathcal{F}_t \subset \mathcal{F}_{t-1} \subset \ldots \subset \mathcal{F}_0 \tag{8}$$

строго упорядоченных по включению диаграмм Ферре (конечных, если идеал I унитарен), и набор

$$0 \leqslant a_0 < a_1 < \ldots < a_t < a_{t+1} = n \tag{9}$$

целых чисел, удовлетворяющие следующим условиям:

(C1) для любых $F \in R[X]$ и $s \in \overline{0, t}$

$$(F \in I, \mathrm{supp}(F) \subseteq \mathcal{F}_s) \implies (\|F\| \geqslant a_{s+1});$$

(C2) для каждого $s \in \overline{0, t}$ существует круллева система $\chi_s \subset R[X]$ с опорной диаграммой Ферре $\mathcal{F}_s$ такая, что I содержит систему полиномов $\pi^{a_s}\chi_s$.

Любая система многочленов

$$\chi_0, \chi_1, \ldots, \chi_t, \tag{10}$$

удовлетворяющая условиям (C1), (C2), обладает также следующими свойствами.

(C3) Если $F \in I$ и $\|F\| = a$, где $a \geqslant a_s$, для некоторого $s \in \overline{0, t}$, то

$$F \in (\pi^a\chi_s \cup \pi^{a_{s+1}}\chi_{s+1} \cup \ldots \cup \pi^{a_t}\chi_t),$$

в частности $I = (\pi^{a_0}\chi_0 \cup \pi^{a_1}\chi_1 \cup \ldots \cup \pi^{a_t}\chi_t)$.

(C4) Если $s \in \overline{0, t}$ и $a_s \leqslant a < a_{s+1}$, то

$$I \cap \pi^a R[X] = (\pi^a\chi_s \cup \pi^{a_{s+1}}\chi_{s+1} \cup \ldots \cup \pi^{a_t}\chi_t), \tag{11}$$

$$(I : \pi^a) = (\chi_s \cup \pi^{a_{s+1}-a}\chi_{s+1} \cup \ldots \cup \pi^{a_t-a}\chi_t \cup \{\pi^{n-a}\}). \tag{12}$$

(C5) Если $H \in R[X]$ *и* $H_1, \ldots, H_{t+1}$ *— многочлены, построенные рекурсивно по правилу:* $H_1 \in \mathrm{Nor}_{\chi_0}(H)$, $H_{s+1} \in \mathrm{Nor}_{\chi_s}(H_s)$, $s \in \overline{1,t}$, *то включение* $H \in I$ *эквивалентно системе условий*

$$\|H_s\| \geqslant a_s, \ s \in \overline{1,t+1}. \tag{13}$$

(C6) Если R *— конечное кольцо и идеал* I *унитарен, то* $S = R[X]/I$ *— конечное кольцо и справедливо равенство (данное равенство для случая* $k = 1$ *было впервые доказано в [1])*

$$|S| = q^{(m_0-m_1)a_1+\ldots+(m_{t-1}-m_t)a_t+m_t n}, \tag{14}$$

где $q = \left|\overline{R}\right|$ *и* $|\mathcal{F}_s| = m_s$ *для* $s \in \overline{0,t}$.

Система полиномов

$$\chi = \pi^{a_0}\chi_0 \cup \pi^{a_1}\chi_1 \cup \ldots \cup \pi^{a_t}\chi_t \tag{15}$$

из теоремы 1 является согласно пункту (C5) стандартным базисом идеала I, согласованным с нормой (2) кольца R. Этот специальный стандартный базис был назван в работах [2, 3, 4, 5] *канонической системой образующих (КСО)* идеала I.

Заметим, что в работе [18] при исследовании *минимального сильного базиса Грёбнера (МСБГ)* из [14] была получена характеризация МСБГ, фактически утверждающая совпадение понятий МСБГ и КСО из [2]. Вместе с тем работа [18] не содержит ссылок на [2]. Также, в работе [19] была предложена формула для вычисления мощности циклического кода C над кольцом Галуа R, по МСБГ, ассоциированного с C идеала $I \lhd R[x]/(x^m - 1)$ (здесь m —длина кода C). Отметим, что формула может быть легко получена, исходя из формулы (14).

В [6] понятие согласованного с нормой стандартного базиса полиномиального идеала изучается с общих позиций в терминах схемы симплификации из [9]. Там же вводится понятие S-полинома, обобщающее аналогичное понятие для многочленов над полем:

Определение 6 ([6], п. 3)**.** Пусть $F, G \in R[X] \setminus 0$ и $\mathrm{Lt}(F) = \alpha\pi^a u$, $\mathrm{Lt}(G) = \beta\pi^b v$, где $a, b \in \overline{0,n-1}$, $u, v \in [X]$ и $\alpha, \beta \in R^*$. Пусть $w = \gcd(u, v) \in [X]$ — наибольший общий делитель мономов u и v, и пусть $c = \max\{a, b\}$. Существуют мономы $u', v' \in [X]$, такие, что $u = wu'$ и $v = wv'$. *S-полиномом* от F и G называется полином

$$S(F, G) = \alpha^{-1}\pi^{c-a}v'F - \beta^{-1}\pi^{c-b}u'G. \tag{16}$$

Также, для удобства, полагаем $S(F, 0) = S(0, G) = 0$.

Следующая, доказанная в [6], теорема является аналогом леммы о композиции (см., например, [14, 15]) из теории базисов Грёбнера над полями.

Теорема 2 ([6], 3.6)**.** *Пусть* χ *— непустая система полиномов из идеала* I *кольца* $R[X]$. *Тогда следующие утверждения эквивалентны:*

(a) $\forall F \in I : \exists G \in \chi \ \mathrm{Lt}(G) \mid \mathrm{Lt}(F)$;

(b) $\forall F \in I : \mathrm{Nor}_\chi(F) = 0$;

(c) $\forall F \in I : \mathrm{Nor}_\chi(F) \ni 0$;

(d) $I = (\chi)$ *и* $\forall G_1, G_2 \in \chi : \mathrm{Nor}_\chi(S(G_1, G_2)) = 0$;

(e) $I = (\chi)$ *и* $\forall G_1, G_2 \in \chi : \mathrm{Nor}_\chi(S(G_1, G_2)) \ni 0$.

В [6] на основе теоремы 2 был представлен алгоритм, формально повторяющий известный алгоритм для полей, позволяющий эффективно вычислять стандартный базис (и в частности КСО) идеала исходя из его системы образующих.

Алгоритм 1 (для вычисления стандартного базиса, [6, 3.9])

INPUT: $\varphi = \{F_1, \ldots, F_s\} \subset R[X]$
OUTPUT: $\chi = \{G_1, \ldots, G_t\}$ — стандартный базис идеала (φ)
INITIALIZATION: $\chi := \psi,\ \mathcal{G} = \{(F_i, F_j) \mid 1 \leqslant i < j \leqslant s\}$
WHILE $\mathcal{G} \neq \varnothing$ **DO**
Выбираем произвольно $(F, G) \in \mathcal{G}$
$\mathcal{G} := \mathcal{G} \setminus \{(F, G)\}$
Вычисляем любой элемент $H \in \mathrm{Nor}_{\mathfrak{G}_\chi}(S(F, G))$
IF $H \neq 0$ **THEN**
$\mathcal{G} := \mathcal{G} \cup \{(U, H) \mid U \in \chi\}$
$\chi := \chi \cup \{H\}$

4 Стандартные базисы и семейства линейных рекуррентных последовательностей

Пусть R — кольцо из заглавия. Функция $\mu : \mathbb{N}_0^k \to R$ называется *k-последовательностью* над R. Совокупность всех k-последовательностей над R обозначается $R^{\langle k \rangle}$. Множество $R^{\langle k \rangle}$ вместе с покомпонентным сложением и умножением на элементы из R, очевидно, является R-модулем. Более того $R^{\langle k \rangle}$ можно превратить в $R[x_1, \ldots, x_k]$-модуль если для последовательности $\mu \in R^{\langle k \rangle}$ и полинома $F = \sum h_{\boldsymbol{i}} \boldsymbol{x}^{\boldsymbol{i}}$ положить

$$(F\mu)(\boldsymbol{z}) = \sum_{\boldsymbol{i}} h_{\boldsymbol{i}} \mu(\boldsymbol{z} + \boldsymbol{i}).$$

Последовательность $\mu \in R^{\langle k \rangle}$ называется *линейной рекуррентной последовательностью (ЛРП)*, если существует набор унитарных полиномов $F_1(x), \ldots, F_k(x) \in R[x]$ таких, что

$$F_s(x_s)\mu = 0,\ s \in \overline{1, k},$$

В этом случае система полиномов $F_1(x_1), \ldots, F_k(x_k)$ называется системой *характеристических* полиномов последовательности μ.

Пусть $I \lhd R[X] = R[x_1, \ldots, x_k]$. Множество последовательностей

$$L_R(I) = \left\{ \mu \in R^{\langle k \rangle} \;\middle|\; I\mu = 0 \right\} \tag{17}$$

называется *ЛРП-семейством*. Ясно, что $L_R(I)$ можно рассматривать как $R[X]$-модуль и как $R[X]/I$-модуль.

Нормой последовательности $\mu \in R^{\langle k \rangle}$ называется число $\|\mu\| = \min\{\|\mu(\boldsymbol{i})\| \mid \boldsymbol{i} \in \mathbb{N}_0^k\}$.

Множеством внутренних углов диаграммы Ферре $\mathcal{F} \subseteq [X]$, согласно [3], называется множество $\mathcal{CF}$ всех максимальных элементов частично упорядоченного множества $(\mathcal{F}, |)$ (здесь, как и ранее, символ $|$ означает отношение делимости).

Следующая теорема была доказана в [2] при $k = 1$ и в [3] для случая произвольного унитарного идеала $I \lhd R[X]$.

Теорема 3 ([2], § 4, 2; [3], 7.8). *Если идеал I имеет КСО вида (15), то модуль $L_R(I)$ содержит систему последовательностей:*

$$\alpha_1^{(t)}, \ldots, \alpha_{c_t}^{(t)}, \pi^{n-a_t}\alpha_1^{(t-1)}, \ldots, \pi^{n-a_t}\alpha_{c_{t-1}}^{(t-1)}, \ldots, \pi^{n-a_1}\alpha_1^{(0)}, \ldots, \pi^{n-a_1}\alpha_{c_0}^{(0)}, \tag{18}$$

таких, что последовательности $\alpha_c^{(s)} \in R^{\langle k \rangle}$ удовлетворяют условиям:

$$\overline{\alpha}_{R[X]}\left(\overline{\alpha}_1^{(s)}, \ldots, \overline{\alpha}_{c_s}^{(s)}\right) = L_{\overline{R}}\left(\overline{J}_s\right), \quad s \in \overline{0, t}. \tag{19}$$

Любая такая система последовательностей обладает также следующим свойством: если $\mu \in L_R(I)$ *и* $\|\mu\| = b$, $n - a_{s+1} \leqslant b < n - a_s$, $s \in \overline{0,t}$, *то*

$$\mu \in {}_{R[X]}\left(\pi^{n-a_{s+1}}\alpha_1^{(s)}, \ldots, \pi^{n-a_{s+1}}\alpha_{c_s}^{(s)}, \ldots, \pi^{n-a_1}\alpha_1^{(0)}, \ldots, \pi^{n-a_1}\alpha_{c_0}^{(0)}\right). \tag{20}$$

В частности, такая система порождает $L_R(I)$ *как* $R[X]$*-модуль.*

Если, кроме того, идеал I *— унитарен, то параметры* $c_0, \ldots, c_t$ *можно выбрать так, чтобы* $c_s \leqslant |\mathcal{CF}_s|$, $s \in \overline{0,t}$.

Отметим, что предлагаемая в теореме 3 система $R[X]$-порождающих ЛРП-семейства $L_R(I)$ (18), может быть построена эффективно исходя из КСО идеала I.

Ввиду теоремы 3, представляет интерес вопрос о цикличности $R[X]$-модуля $L_R(I)$. В [2] этот вопрос изучался для случая $k = 1$.

В случае $k = 1$ флаг диаграм Ферре (8) из теоремы 1 есть последовательность вложенных «отрезков»:

$$\mathcal{F}_s = \{1, x, \ldots x^{m_s-1}\}, \quad s \in \overline{0,t}; \quad m_0 > m_1 > \ldots > m_t \geqslant 0, \tag{21}$$

а каждая круллева система χ_s (см. (5)) состоит из одного унитарного полинома F_s степени m_s, $s \in \overline{0,t}$. В этом случае КСО идеала I имеет вид

$$\chi = \{\pi^{a_0}F_0, \ldots, \pi^{a_t}F_t\}. \tag{22}$$

Предложение 4 ([2], § 3, 14). *Идеал* $I \lhd R[x]$ *с КСО (22) является примарным тогда и только тогда, когда либо* $a_0 = 0$ *и* $\overline{F_0}$ *— примарный полином в* $\overline{R}[x]$, *либо* $m_0 = 0$ *(т. е.* $I = (\pi^{a_0})$*).*

Известно (см., например, [12], VI, § 5), что всякий унитарный идеал $I \lhd R[X]$ есть пересечение конечного числа примарных попарно взаимно простых идеалов (называемых *примарными компонентами* идеала I) — *примарное разложение* идеала I.

В случае $k = 1$, как показывает следующая теорема, примарное разложение идеала $I \lhd R[x]$ может быть построено эффективно.

Предложение 5 ([2], § 3, 15). *Пусть унитарный идеал* $I \lhd R[X]$ *имеет КСО (22) (с* $a_0 = 0$*) и пусть* F_0 *есть произведение взаимно простых полиномов:*

$$F_0 = K_0H_0, \quad \gcd(K_0, H_0) = 1.$$

Тогда для каждого $s \in \overline{1,t}$ *имеем* $F_s = K_sH_s$, *где* K_s *и* H_s *— унитарные полиномы со свойствами*

$$\overline{K_s} \mid \overline{K_0}, \quad \overline{H_s} \mid \overline{H_0}, \quad \gcd(K_s, H_s) = 1.$$

При этом справедливо равенство $I = \mathcal{K} \cap \mathcal{H}$, *где*

$$\mathcal{K} = (K_0, \pi^{a_1}K_1, \ldots, \pi^{a_t}K_t), \quad \mathcal{H} = (H_0, \pi^{a_1}H_1, \ldots, \pi^{a_t}H_t).$$

КСО идеала $\mathcal{K}$ *получается из системы*

$$\{K_0, \pi^{a_1}K_1, \ldots, \pi^{a_t}K_t\}$$

вычеркиванием каждого полинома $\pi^{a_s}K_s$, *удовлетворяющего условию* $\deg K_s = \deg K_{s-1}$.

Пусть $I = I_1 \cap \ldots \cap I_r$ есть примарное разложение унитарного идеала I. Согласно [2], § 1, 6, ЛРП-семейство $L_R(I)$ циклично, в том и только том случае, когда все семейства $L_R(I_s)$, $s \in \overline{1,r}$ цикличны. Таким образом, предложение 5 сводит общий вопрос о цикличности ЛРП-семейства, к вопросу о цикличности для унитарного примарного идеала I.

Теорема 4 ([2], § 5, 3). *Пусть* I *— унитарный примарный идеал с КСО (22) (с* $a_0 = 0$*). Определим системы полиномов* $\{Q_{ij} \mid 0 \leqslant i < j \leqslant t\}$ *и* $\{B_{ij} \mid 0 \leqslant i < j \leqslant t\}$ *равенством*

$$F_i = Q_{ii+1}F_{i+1} - \pi^{a_{i+2}-a_{i+1}}Q_{ii+2}F_{i+2} - \ldots - \pi^{a_j-a_{i+1}}Q_{ij}F_j - \pi^{a_{j+1}-a_{i+1}}B_{ij},$$

где $\deg Q_{is}F_s < m_{s-1}$ *для* $s \in \overline{i+2,j}$ *и* $\deg B_{ij} < m_j$. *Тогда* $R[x]$*-модуль* $L_R(I)$ *является циклическим в том и только том случае, если либо* $t = 0$ *(т. е.* I *— главный идеал), либо* $t > 0$ *и*

$$\gcd\left(\overline{B}_{t-1,t}, \overline{F}_t\right) = 1,$$
$$\gcd\left(\overline{Q}_{ii+2}, \overline{F}_{i+1}\right) = 1 \text{ для } i \in \overline{0,t-2}.$$

Итак, предложение 5 и теорема 4 дают алгоритм проверки цикличности семейства $L_R(I)$ по КСО идеала I.

Модуль ${}_RM$ называется *квазифробениусовым* (*QF-модулем*), если для всех $J \lhd R$ и $K < {}_RM$ выполняются следующие соотношения $\mathrm{An}_R(\mathrm{An}_M(J)) = J$ и $\mathrm{An}_M(\mathrm{An}_R(K)) = K$. Для коммутативного артинового кольца R существует единственный с точностью до изоморфизма QF-модуль ${}_RQ$, причем $(Q,+) \cong (R,+)$. Кольцо R называется *квазифробениусовым (QF-кольцом)*, если ${}_RR$ является QF-модулем.

Следующая теорема дает критерий цикличности ЛРП-семейства для произвольного унитарного идеала кольца полиномов $R[X] = R[x_1, \ldots, x_k]$ над коммутативным артиновым кольцом R.

Теорема 5 ([1, 7, 8]). *Пусть R — коммутативное артиново кольцо, ${}_RQ$ — QF-модуль и $I \subseteq \mathcal{R}_k$ — унитарный идеал. Тогда семейство $L_Q(I)$ есть QF-модуль над коммутативным артиновым кольцом $S = R[X]/I$ и следующие условия эквивалентны:*

(a) $L_Q(I)$ — *циклический $R[X]$-модуль;*

(b) $I = \mathrm{An}(\mu)$ *для некоторой рекурренты μ;*

(c) S — *квазифробениусово кольцо;*

(d) $\overline{R}\left(R[X]/\sqrt{I}\right) \cong \overline{R}\left(\left(I : \sqrt{I}\right)/I\right)$.

Здесь $\sqrt{I} = \{F \in R[X] \mid \exists\, m \in \mathbb{N} : F^m \in I\}$ — *радикал* идеала I.

Отметим, что эквивалентность условий (b) и (d), в случае если $k = 1$ и R — локальное кольцо главных идеалов (т. е. $Q = R$) или когда k — произвольно и R — поле, была получена ранее в [16, 17]. Вместе с тем, доказательство теоремы 5 намного короче чем доказательство соответствующих результатов в [16, 17], так как использует хорошо известные свойства QF-колец.

Отметим, что в рассматриваемом в теореме 5 общем случае не было найдено алгоритма распознающего цикличность ЛРП-семейства. Таким образом результаты теоремы 4 не покрываются теоремой 5. Задача отыскания указанного алгоритма в общем случае остается открытой.

5 Стандартные базисы и системы полиномиальных уравнений

Пусть $R = GR(q^n, p^n)$ — кольцо Галуа характеристики p^n и порядка q^n. Кольцо R является коммутативным артиновым цепным кольцом с единственным максимальным идеалом pR индекс нильпотентности которого равен n, $\overline{R} = R/pR$ — поле Галуа $GF(q)$. Напомним, что каждый элемент $a \in R$ однозначно представляется в виде:

$$a = a^{(0)} + a^{(1)}p + \ldots + a^{(n-1)}p^{n-1}, \tag{23}$$

где $a^{(0)}, \ldots, a^{(n-1)} \in \Gamma = \{\alpha \in R \;:\; \alpha^q = \alpha\}$. Множество Γ называется *координатным полем (полем Тейхмюллера)* кольца R. Разложение (23) называется *p-адическим разложением*, элемент $a^{(i)}$, $i \in \overline{0, n-1}$, называется i-ой *координатой элемента a*. Для произвольных $j \in \overline{0, n-1}$ и $\boldsymbol{c} = (c_1, \ldots, c_k) \in R^k$ вектор $\boldsymbol{c}^{(j)} = (c_1^{(j)}, \ldots, c_k^{(j)}) \in \Gamma^k$ будем называть, как и в [4, 5], *j-ым координат-вектором вектора $\boldsymbol{c}$*.

Рассмотрим систему уравнений над R:

$$\begin{cases} F_1(x_1, \ldots, x_k) = 0 \\ \qquad\qquad \ldots \\ F_d(x_1, \ldots, x_k) = 0 \end{cases} \tag{24}$$

где $F_1, \ldots, F_d \in R[X]$.

В [4, 5] излагается способ решения системы уравнений (24) путем последовательного определения координат p-адического разложения каждого решения (в случае, когда $R = \mathbb{Z}_{p^n}$, подобные системы изучались ранее (см., например, [13])).

Положим $\psi = \{F_1, \ldots, F_d\}$ и будем коротко записывать систему (24) в виде

$$\psi(\boldsymbol{x}) = 0. \tag{25}$$

Суть обсуждаемого метода состоит в том, что множество K решений системы (25) строится путем решения систем сравнений

$$\psi(\boldsymbol{x}) \equiv 0 \pmod{p^j}, \tag{26}$$

последовательно для $j \in \overline{1,n}$. Сначала находятся все принадлежащие Γ^k решения системы сравнений

$$\psi(\boldsymbol{z}) \equiv 0 \pmod{p}. \tag{27}$$

Затем, если уже найдены решения $\boldsymbol{c}^{[j]} \in \Gamma^k + p\Gamma^k + \ldots + p^{j-1}\Gamma^k$ системы (26), для каждого из них строится множество решений $\boldsymbol{c}^{[j]} + p^j\boldsymbol{c}^{(j)} \in \Gamma^k + p\Gamma^k + \ldots + p^j\Gamma^k$ системы сравнений

$$\psi(\boldsymbol{x}) \equiv 0 \pmod{p^{j+1}}, \tag{28}$$

при этом, координат-вектор $\boldsymbol{c}^{(j)} \in \Gamma^k$ находится как решение системы сравнений

$$\psi(\boldsymbol{c}^{[j]} + p^j\boldsymbol{z}) \equiv 0 \pmod{p^{j+1}}. \tag{29}$$

Как показано в [4, 5], система сравнений (29) равносильна линейной системе

$$D\psi(\boldsymbol{c}^{[j]}) \cdot \boldsymbol{z}^{\downarrow} \equiv -\psi^{\downarrow}(\boldsymbol{c}^{[j]})^{(j)} \pmod{p}, \tag{30}$$

здесь $\psi(\boldsymbol{x}) = (D_sF_i(\boldsymbol{x}))_{d\times k}$ (D_s — оператор дифференцирования по x_s) и $\psi^{\downarrow}(\boldsymbol{x}) = (F_1(\boldsymbol{x}), \ldots, F_d(\boldsymbol{x}))^T$.

Рассмотренный выше подход реализуется следующим алгоритмом решения системы (25):

Алгоритм 2 (решения системы уравнений (25), [4, 5])

Вход: система уравнений (25).

Выход: множество K всех решений в R^k системы (25).

Алгоритм состоит в последовательном заполнении двумерных массивов (таблиц) A_0, A_1, ..., A_{n-1} элементами из Γ^k. Количество столбцов в j-м массиве равно $j+1$, а количество строк меняется в процессе работы алгоритма. В начале работы алгоритма все массивы пусты. Алгоритм заканчивает работу, если либо для некоторого $j < n-1$ массив A_j после его заполнения остался пустым (в таком случае система не имеет решений), либо завершено заполнение массива A_{n-1} (в таком случае множество решений K есть множество строк массива A_{n-1}).

Этап I. Найдем все решения в Γ^k системы сравнений 27 (используя, например, полный перебор). Пусть $C^{(0)} \subset \Gamma^k$ — множество решений этой системы.

Если $C^{(0)} = \varnothing$, то $K = \varnothing$ и работа алгоритма закончена.

В противном случае, перенумеровав каким-либо образом все получившиеся решения, поместим их в массив A_0. Если $n = 1$, то $K = C^{(0)}$ и алгоритм завершен. Если $n > 1$, то обозначаем через b_0 число строк в массиве A_0 и переходим ко второму этапу.

Этап II. Процедура подъема $\boldsymbol{c}^{(0)}$. Этот этап состоит из $n-1$ шагов. На j-ом шаге, $j \in \overline{1,n-1}$, строится массив A_j, строки которого формируются путем последовательной обработки строк уже построенного массива A_{j-1}. Пусть $b_j(v)$ — число строк массива A_j, полученных после обработки первых v строк массива A_{j-1}, и b_j — общее число строк массива A_j ($b_j = b_j(b_{j-1})$).

Пусть $v \in \overline{1,b_{j-1}}$ и в данный момент обрабатывается v-я строка массива A_{j-1}: $A_{j-1}(v,1) = \boldsymbol{c}^{(0)}$, $A_{j-1}(v,2) = \boldsymbol{c}^{(1)}$, ..., $A_{j-1}(v,j) = \boldsymbol{c}^{(j-1)}$. Строки массива A_{j-1} выбираются последовательно, так что строки с номерами $\overline{1,v-1}$ уже рассмотрены и в массив A_j записано $b_j(v-1)$ строк.

Пусть $C^{(j)} = \{\boldsymbol{c}^{(0)}, \boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(j-1)}\} \subset \Gamma^k$ — множество решений линейной системы (30) и пусть $l = |C^{(j)}|$. Полагаем $b_j(v) = b_j(v-1) + l$. Если $l > 0$, то, занумеровав произвольным образом элементы множества $C^{(j)}$: $C^{(j)} = \{\boldsymbol{c}_1^{(j)}, \ldots, \boldsymbol{c}_l^{(j)}\}$, заполняем массив A_j: $A_j(b_j+i,1) = \boldsymbol{c}^{(0)}$, $A_j(b_j+i,2) = \boldsymbol{c}^{(1)}$, ..., $A_j(b_j+i,j) = \boldsymbol{c}^{(j-1)}$, $A_j(b_j+i,j+1) = \boldsymbol{c}_i^{(j)}$.

Если $v = b_{j-1}$, то процедура формирования массива A_j закончена, иначе рассматриваем $(v+1)$-ую строку массива A_{j-1}.

Если $b_j = 0$, т. е. массив A_j пуст, то алгоритм завершает работу, в противном случае переходим к построению массива A_{j+1}.

По завершении формирования массива A_{n-1} полагаем

$$K = \left\{A_{n-1}(i,1) + pA_{n-1}(i,2) + \ldots + p^{n-1}A_{n-1}(i,n) : i \in \overline{1,b_{n-1}}\right\} \subset R^k.$$

Предложение 6 ([4, 5]). *Алгоритм 2 за конечное число шагов построит множество решений системы уравнений (25).*

Пусть χ — КСО идеала $(\psi) \lhd R[X]$ представленная в виде (15). Ясно, что система уравнений (25) равносильна системе

$$\begin{cases} \psi_0(\boldsymbol{x}) = 0 \\ p\psi_1(\boldsymbol{x}) = 0 \\ \dots \\ p^{n-1}\psi_{n-1}(\boldsymbol{x}) = 0, \end{cases} \tag{31}$$

где $\psi_j = \chi_i$, если $a_i \leqslant j < a_{i+1}$.

Теорема 6 ([4, 5]). *Вектор $\boldsymbol{c} \in R^k$ есть решение системы уравнений (31) тогда и только тогда, когда*

(a) *нулевой координатный вектор $\boldsymbol{c}^{(0)} \in \Gamma^k$ есть решение системы уравнений*

$$\psi_{n-1}(\boldsymbol{z}) \equiv 0 \pmod{p}, \tag{32}$$

(b) *для каждого $j \in \overline{1, n-1}$ координатный вектор $\boldsymbol{c}^{(j)} \in \Gamma^k$ есть решение системы*

$$\psi_{n-j-1}\left(\boldsymbol{c}^{[j]} + p^j \boldsymbol{z}\right) \equiv 0 \pmod{p^{j+1}}. \tag{33}$$

На основании теоремы 6, в работах [4, 5] был построен *модифицированный алгоритм* решения систем вида (31).

В модифицированном алгоритме на первом этапе находится множество решений в Γ^k системы (32), а затем, с помощью соотношений (33) производится их подъем. Здесь, как и в алгоритме 2, процедура подъема сводится к решению систем линейных уравнений

$$D\psi^{(0)}_{n-j-1}(\boldsymbol{c}^{(0)}) \cdot \boldsymbol{z}^{\downarrow} = -\psi^{\downarrow}_{n-j-1}(\boldsymbol{c}^{[j]})^{(j)}, \quad j \in \overline{1, n-1}. \tag{34}$$

При подъеме нулевой координаты, с помощью модифицированного алгоритма, решается система линейных уравнений над Γ^k не со всей КСО χ в левой части (как в алгоритме 2), а лишь с какой-то одной ее компонентой — χ_i. Поэтому, в применении к системе уравнений вида (31), модифицированный алгоритм эффективнее алгоритма 2.

Алгоритм 2 на первом этапе находит корни системы

$$\psi_0(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Gamma^k, \tag{35}$$

в то время как модифицированный алгоритм решает систему

$$\psi_{n-1}(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Gamma^k. \tag{36}$$

Из соотношений (8) следует, что система (36) имеет, в некотором смысле, меньшую «степень» чем система (35) — в этом состоит еще одна причина большей эффективности модифицированного алгоритма.

Рассмотрим, например, кольцо $\mathbb{Z}_{2^n}[x_1, \dots, x_k]$ и систему уравнений с левой частью (являющейся КСО) $\chi = \chi_0 \cup 2^{n-1}\chi_1$, где

$$\chi_0 = \{x_1^2 - x_1, \dots, x_k^2 - x_k\} \text{ и } \chi_1 = \{1\}. \tag{37}$$

Система уравнений (31) не имеет решений над $\mathbb{Z}_{2^n}$. При использовании алгоритма 2 потребуется найти корни системы $\chi_0(\boldsymbol{x}) = 0$ в поле $\mathbb{Z}_2$, множество которых, как легко видеть, есть $\mathbb{Z}_2^k$, вместе с тем, модифицированный алгоритм завершится на первом шаге, поскольку в нашем случае $\psi_{n-1} = \chi_1 = \{1\}$.

Литература

[1] Kurakin V. L., Kuzmin A. S., Mikhalev A. V., Nechaev A. A. Linear recurring sequences over rings and modules. (Contemporary Math. and its Appl. Thematic surveys. Vol. 10. Algebra 2. Moscow, 1994.) J. of Math. Sciences, **76** (1995), № 6, 2793–2915.

[2] Нечаев А. А. Линейные рекуррентные последовательности над коммутативными кольцами. Дискретная Математика, т. **3**, № 4 (1991) с. 105–127.

[3] Нечаев А. А., Михайлов Д. А. *Каноническая система образующих унитарного полиномиального идеала над коммутативным артиновым цепным кольцом.* Дискретная математика, т. **13**, № 4 (2001) с. 3–42.

[4] Нечаев А. А., Михайлов Д. А. Использование КСО для решения систем уравнений над кольцами. 22-я межведомственная научно-техн. конф., Серпуховский ВИРВ (2003) с. 49–52.

[5] Михайлов Д. А., Нечаев А. А. Решение систем полиномиальных уравнений над кольцами Галуа с помощью канонической системы образующих полиномиального идеала. Дискретная математика, 2004 (в печати).

[6] Горбатов Е. В. Стандартный базис полиномиального идеала над коммутативным артиновым цепным кольцом. Дискретная математика, (2003) (в печати).

[7] Nechaev A. A. Polylinear recurring sequences over modules and quasi-Frobenius modules. Proc. First Int. Tainan–Moscow Algebra Worcshop, 1994, Walter de Gruyter, Berlin–N. Y. (1996), pp. 283–298.

[8] Горбатов Е. В., Нечаев А. А. Критерий цикличности семейства полилинейных рекуррент над QF-модулем. Успехи мат. наук, **56** (2001) № 4, с. 167–168.

[9] Латышев В. Н. Комбинаторная теория колец, стандартные базисы, М.: Изд-во МГУ, 1988.

[10] Kronecker L. Vorlesungen über Zahlentheorie. Bd. **1**, Leipzig, Teubner, 1901.

[11] Фейс К. Алгебра: кольца, модули и категории, т. 2. М.: Мир, 1979.

[12] Ленг С. Алгебра, М.: Мир, 1968.

[13] Бухштаб А. А. Теория чисел. Пер. с англ. М.: Просвещение, 1966.

[14] Adams W., Loustaunau P. An Introduction to Gröbner bases. Graduate Studies in Mathematics, v. **3**, American Mathematical Society, 1994.

[15] Кокс Д., Литтл Дж., О'Ши Д. Идеалы, многообразия и алгоритмы. М.: Мир, 2000.

[16] Lu P. Z., Liu M. L. A formula to determine zero-dimensional ideals being annihilating ideals of LRAs. Algebra Colloquium, 6:3, 1999, pp. 349–360.

[17] Lu P. Z. A Criterion for annihilating ideals of linear recurring sequences over galois rings. AAECC vol. **11**, No. 2, 2000.

[18] Norton G. H., Salagean A. Strong Gröbner bases and cyclic codes over a finite-chain ring. Proceedings of the International Workshop on Coding and Cryptography, Paris, 2001, Jan., pp. 8–12.

[19] Norton G. H., Salagean A. On the structure of linear and cyclic codes over a finite chain ring. Applicable Algebra in Engineering, Communication and Computing, 2000, **10**, № 6, pp. 489–506.

[20] Byrne E., Fitzpatrick P. Gröbner bases over Galois rings with an application to decoding alternant codes. J. Symbolic Computation, v. **31**, 2001, pp. 565–584.

Некоторые свойства больших простых делителей чисел вида $p-1$

М. А. Черепнёв

При использовании задачи дискретного логарифмирования для построения криптосхем, довольно часто она трансформируется в задачу Диффи-Хеллмана [1]. Стойкость построенных таким способом схем сводится к вычислительной сложности решения задачи Диффи-Хеллмана. Является ли эта задача в общем случае более простой, чем задача дискретного логарифмирования - вопрос пока открытый. В некоторых специальных случаях [2, 3] удаётся доказать их вычислительную полиномиальную эквивалентность с помощью построения алгоритмов дискретного логарифмирования с оракулом Диффи-Хеллмана. В общем случае такой алгоритм построен в работе [4]. Для оценки скорости его работы требуется оценка следующей теоретико-числовой функции.

Пусть

$$m = \prod_{i=1}^{r} {p_i}^{\alpha_i}.$$

Рассмотрим разложение на простые множители чисел $p_i - 1, i = 1, \ldots, r$. Из каждого из этих простых вычтем единицу и снова разложим на простые, и так далее. Получившееся разветвление назовём деревом числа m, а встречающиеся в нём простые числа его узлами. Обозначим $s = s(m)$ длину наибольшей ветви дерева числа m.

Обозначим $L(t, m)$ - количество битовых операций, необходимых для решения задачи дискретного логарифмирования в группе порядка m и с групповой операцией битовой сложности t. Пусть $D(t, m)$ - количество битовых операций, необходимых для решения задачи Диффи-Хеллмана в такой же группе.

В работе [4] доказана следующая теорема.

Теорема 1. *Для сертифицированной задачи дискретного логарифмирования*

$$L(t, m) \leqslant s \cdot \log^2 m \cdot D(\ldots D(D(t, m), m) \ldots),$$

где справа стоит s-кратная итерация функции $D(t, m)$. (Здесь, как и дальше, не указанное основание логарифма есть некоторая подходящая константа.)

Следствие. *Для алгоритмов, скорость работы которых удовлетворяет неравенству*

$$D^*(t, m) \leqslant tD^*(C, m), \tag{1}$$

где C - некоторая константа, выполнено

$$L(t, m) \leqslant t \cdot s \cdot \log^2 m \cdot (D^*(C, m))^s.$$

Таким образом, при

$$\frac{s(m)}{\log m} \log\log m$$

стремящемся к нулю при m стремящемся к бесконечности, полиномиальный алгоритм решения задачи Диффи-Хеллмана, скорость работы которого удовлетворяет (1) для некоторой группы, даёт в ней же алгоритм решения задачи дискретного логарифмирования, работающий быстрее перебора.

Для оценки $s(m)$ для почти всех m можно применить результаты о количестве различных простых делителей чисел вида $p-1$. Оценка снизу для этого количества даёт оценку сверху для самих простых делителей, из которой можно в дальнейшем получить оценку для $s(m)$.

В своих работах Харди и Рамануджан [5], Туран [6] показали, что за исключением $\overline{o}(x)$ натуральных значений $n, n \leqslant x$ выполняются неравенства

$$(1 - \varepsilon) \ln\ln n < v(n) < (1 + \varepsilon) \ln\ln n,$$

где $v(n)$ - количество различных простых делителей числа n. Аналогичный результат для чисел вида $p-1$, где p - простое число, получил Эрдёш [7]. В данной заметке похожий результат получен для количества простых делителей чисел $q-1$ для больших простых q, $q|p-1$. Заметим, что ввиду очевидной оценки $s(n) \leqslant \log n$ маленькие простые делители не оказывают влияния на величину $s(m)$.

Пусть $N(M)$ - количество элементов в множестве M.

Лемма. *Пусть* $k, n \in \mathbb{N}$. *Тогда найдутся такие абсолютные положительные константы* c_1, c_2, *что*

$$N\left(p \leqslant x \;\middle|\; \frac{p-1}{n} = q \text{ простое}, v(q-1) = k\right) < c_1 \frac{x(n+1)(\ln\ln\frac{x}{n} + c_2)^{k+2}}{(k-1)!\varphi(n(n+1))\ln^3\frac{x}{n}} + \overline{o}\left(\frac{x}{n\ln\frac{x}{n}^{\frac{1}{2}}\ln\ln\ln\frac{x}{n}}\right),$$

причём, $\overline{o}$ *не зависит от* k *и от* n.

Из этой леммы суммированием по k и по n получена следующая

Теорема 2. *Для любого положительного* ε

$$N\left(p \leqslant x \;\middle|\; \text{для любого простого } q : p-1 = qn,\ n \in \mathbb{N},\ q > e^{\ln x/\ln\ln x} \text{ выполнено } v(q-1) \in \left[(1-\varepsilon)\ln\ln\frac{x}{n},\ (1+\varepsilon)\ln\ln\frac{x}{n}\right]\right) = \frac{x}{\ln x} + \overline{o}\left(\frac{x}{\ln x}\right).$$

Доказательства опираются на известную теорему из метода решета Сельберга (Теорема 2.4.6 [8]) о количестве простых чисел, для которых линейные комбинации с фиксированными целыми коэффициентами тоже являются простыми.

Заметим, что теми же средствами могут быть получены аналогичные результаты для всех простых чисел из дерева числа n, отстоящих от n на постоянное, то есть не растущее с ростом n, число звеньев. Для получения указанным способом таких оценок для остальных узлов дерева n необходимо в методе Сельберга получить более точную и эффективную зависимость от количества указанных линейных комбинаций.

Литература

[1] Sacurai K., Shizuya H. Relationships among the Computational Powers of Breaking Discrete Log Cryptosystems. Eurocrypt'95, 341–355.

[2] Maurer U. M. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms. Crypto'94, 271–281.

[3] Boer B. Diffie-Hellman is as strong as discrete log for certain primes. Lect. Notes. Comp. Sci, 1988, 403, 530–540.

[4] Черепнёв М. А. О связи сложностей задач дискретного логарифмирования и Диффи-Хеллмана. Дискр. мат., с. 22–30.

[5] Hardy G. H., Ramanujan S. The normal number of prime factors of a number n. Quart. J. of Math., 1917, 48, 76–92.

[6] Turán P. On a theorem of Hardy and Ramanujan. J. Lond. Math. Soc. (2), 1929, 30, 93–111.

[7] Erdős P. On the normal number of prime factors of $p-1$ and some related problems concerning Euler's φ-function. Quart. J. Oxford, 1935, 6, 205–213.

[8] Прахар К. Распределение простых чисел. М.: Мир, 1967, 511 с.

О групповых свойствах одной системы уравнений

А. Ю. Нестеренко

Пусть $\mathbb{K}$ поле характеристики, отличной от 2 и $\overline{\mathbb{K}}$ его алгебраическое замыкание. Рассмотрим систему уравнений

$$\begin{cases} a^2x_1^2 = c^2x_0^2 - b^2x_2^2, \\ a^2x_0^2 = b^2x_3^2 + c^2x_1^2, \end{cases} \tag{1}$$

где $a, b, c \in \overline{\mathbb{K}}$, отличны от нуля и удовлетворяют соотношению

$$c^4 = a^4 + b^4.$$

Будем считать, что решения системы принадлежат проективному пространству $\mathbb{P}^3(\overline{\mathbb{K}})$. Таким образом определяется эллиптическая кривая над полем $\mathbb{K}$.

Введем на множестве решений системы (1) операцию сложения. Пусть (x_0, x_1, x_2, x_3) и (y_0, y_1, y_2, y_3) два решения. Тогда определим

$$\begin{aligned} z_0 &= bc(x_0x_1y_2y_3 - x_2x_3y_0y_1), \\ z_1 &= ab(x_0^2y_3^2 - x_3^2y_0^2), \\ z_2 &= c^2(x_1x_2y_0y_3 - x_0x_3y_1y_2), \\ z_4 &= ac(x_1x_3y_0y_2 - x_0x_2y_1y_3). \end{aligned} \tag{2}$$

если правые части (2) одновременно не равны нулю. В противном случае, определим

$$\begin{aligned} z_0 &= ac(x_1x_2y_1y_2 + x_0x_3y_0y_3), \\ z_1 &= a^2(x_0x_2y_1y_3 + x_1x_3y_0y_2), \\ z_2 &= ab(x_2x_3y_2y_3 + x_0x_1y_0y_1), \\ z_3 &= bc(x_2^2y_2^2 - x_0^2y_0^2). \end{aligned} \tag{3}$$

Можно непосредственно проверить, что значения (z_0, z_1, z_2, z_3) также будут являться решением системы (1). Применение формул (3) необходимо в случае выполнения равенств $x_i = \varepsilon_i y_i$, $\varepsilon_i = \pm 1$, $i = 0, \ldots, 3$ и $\varepsilon_0 = \varepsilon_1\varepsilon_2\varepsilon_3$.

Введенная операция позволяет задать на множестве решений структуру абелевой группы. Коммутативность операции следует из формул (2) и (3), а тождества, подтверждающие ассоциативность следуют из теорем сложения для тета-функций (см. [2]). Кроме того, для произвольного решения (x_0, x_1, x_2, x_3), получаем равенства

$$\begin{aligned} (x_0, x_1, x_2, x_3) + (b, 0, c, a) &= (x_0, x_1, x_2, x_3), \\ (x_0, x_1, x_2, x_3) + (x_0, -x_1, x_2, x_3) &= (b, 0, c, a), \end{aligned}$$

которые определяют нулевой и обратный элементы в группе решений системы (1).

Теперь мы покажем как система (1) может быть использована в некоторых математических приложениях. Мы нормируем решения системы (1) и перейдем к аффинным координатам. Пусть

$$t_1 = \frac{ax_1}{cx_0} \quad t_2 = \frac{bx_2}{cx_0}, \quad t_3 = \frac{bx_3}{ax_0} \quad \text{и} \quad \lambda = \frac{c^4}{a^4}.$$

Тогда система (1) принимает вид

$$\begin{cases} t_1^2 + t_2^2 = 1, \\ \lambda t_1^2 + t_3^2 = 1. \end{cases} \tag{4}$$

При этом равенства (2) и (3) позволяют определить для полученной системы законы сложения, совпадающие с формулами сложения эллиптических функций Якоби (см. [1]). Как показано автором, полученная система, при вычислении в однородных координатах, имеет меньшую трудоемкость операции сложения, чем на эллиптической кривой в форме Вейерштрасса и может использоваться в практических приложениях, критичных по времени исполнения.

Далее, в случае $\overline{\mathbb{K}} = \mathbb{C}$ определим $\varkappa$ равенством $\varkappa^2 = -a^4$ и сделаем еще одну замену переменных

$$z_1 = \frac{t_1}{\varkappa} = -\frac{ix_1}{acx_0}, \quad z_2 = t_2 = \frac{bx_2}{cx_0}, \quad z_3 = t_3 = \frac{bx_3}{ax_0},$$

где $i^2 = -1$. Тогда система (4) принимает вид

$$\begin{cases} z_2^2 - uz_1^2 = 1, \\ z_3^2 - vz_1^2 = 1. \end{cases} \tag{5}$$

Подобные системы возникают при решении ряда диофантовых задач, и существование групповой структуры на множестве их решений может быть использовано для нахождения решений системы (5).

Литература

[1] Гурвиц А., Курант Р. Теория функций, М: Наука, 1968.

[2] Мамфорд Д.Лекции о тета-функциях. НФМИ, 1998.

Генерация неприводимых многочленов данной степени

О. Е. Демкина, А. В. Торгашова

Для генерации «криптографических» многочленов [1, 2, 3, 4, 5] можно предложить [6] метод последовательного определения неприводимых многочленов, корни которых связаны степенным соотношением. Пусть $f(x)$ – неприводимый многочлен

$$f(x) = x^n + f_{n-1}x^{n-1} + \ldots + f_1x^1 + 1$$

степени n над полем Галуа $\mathbb{Z}_2$, α – его корень, $\alpha \in GF(2^n)$, $f(\alpha) = 0$. Пусть $\beta = \alpha^p$, $p \in \mathbb{N}$, $g(x)$ – характеристический многочлен элемента β, $g(\beta) = 0$,

$$g(x) = x^n + g_{n-1}x^{n-1} + \ldots + g_1x^1 + 1.$$

Поставим задачу определения коэффициентов многочлена g по данному числу p и по коэффициентам многочлена f. Для простоты можно предполагать, что p – простое число.

По теореме 3.39 в [4] для поля характеристики два имеем

$$g(x^p) = \prod_{j=1}^{t} f(\omega_j x), \tag{1}$$

где $\omega_1, \ldots, \omega_p$ – все корни степени p из единицы над $\mathbb{Z}_2$ (т. е. все корни с учетом их кратности).

Пусть ε – первообразный корень степени p из единицы. Можно считать корни $\omega_1, \ldots, \omega_p$ занумерованными так, что $\omega_k = \varepsilon^k$, $k = 0, 1, \ldots, p-1$, $\omega_p = \omega_0 = 1$.

Подставляя эти корни в (1), раскрывая скобки и приводя подобные, получим соотношения

$$g_j = \sum_{k_0+k_1+\ldots+k_{p-1}=pj} f_{k_0}f_{k_1}\ldots f_{k_{p-1}}\varepsilon^{0\cdot k_0+1\cdot k_1+\ldots+(p-1)\cdot k_{p-1}} \tag{2}$$

для всех $j \geqslant 0$ (здесь $0 \leqslant k_0, k_1, \ldots, k_{p-1} \leqslant n$).

Уравнение (2) для j-ого коэффициента многочлена $g(x)$, пользуясь коммутативностью и ассоциативностью умножения, запишем в виде

$$g_j = \sum_{k=1}^{p} \sum f_{m_1}^{n_1} \ldots f_{m_k}^{n_k} \cdot \delta \left(\frac{n_1, \ldots, n_k}{m_1, \ldots, m_k} \right), \tag{3}$$

где вторая сумма распространена на все такие наборы целых неотрицательных чисел $m_1, \ldots, m_k$ и натуральных чисел $n_1, \ldots, n_k$, что $0 \leqslant m_1 < \ldots < m_k \leqslant n$, $n_1 + \ldots + n_k = p$, $m_1 n_1 + \ldots + m_k n_k = pj$, а величина δ дается выражением

$$\delta \left(\frac{n_1, \ldots, n_k}{m_1, \ldots, m_k} \right) = \sum \varepsilon^{0 \cdot k_0 + 1 \cdot k_1 + \ldots + (p-1) \cdot k_{p-1}} \tag{4}$$

где сумма – по всем таким p-мерным векторам $\overline{k} = (k_0, k_1, \ldots, k_{p-1})$ (из целых неотрицательных чисел $k_0, k_1, \ldots, k_{p-1}$), в которых ровно n_1 координат равны m_1, ровно n_2 координат равны m_2, и т. д., ..., ровно n_k координат равны m_k. По указанной выше теореме величина δ может принимать значения только нуль и единица. Если $\delta = 0$, то одночлен $f_{m_1}^{n_1} \ldots f_{m_k}^{n_k}$ не входит в выражение для g_j, а если $\delta = 1$, то этот одночлен входит в g_j. Задача свелась к определению величины δ в зависимости от наборов $m_1, \ldots, m_k$ и $n_1, \ldots, n_k$.

Рассмотрение конкретных значений p приводит к простым явным рекуррентным формулам, легко реализуемым программно. Эти формулы выписаны в аналитическом виде для малых $p = 3, 5$. При непростом p можно использовать их комбинации, например при $p = 9$ применяем два перехода $\alpha \to \alpha^3$, так как $\alpha^9 = (\alpha^3)^3$.

Так [6], при преобразовании $x \to x^3$ имеем как следствия теоремы 3.39 в [4] рекуррентные формулы, непосредственно выражающие коэффициенты b_j функции $g(x)$ через коэффициенты a_k функции $f(x)$:

$$b_j = \sum_{m+k+l=3j}^{\infty} a_m a_k a_l,$$

где данная сумма распространена на все такие множества $\{m, k, l\}$ индексов, что $0 \leqslant m, k, l \leqslant n$, $m + k + l = 3j$, причем такие, что в том случае, когда все m, k, l различны, не должны одновременно выполняться сравнения $m \equiv k \equiv \pmod 3$.

При $p = 5$ получаем, что коэффициент g_j равен сумме следующих слагаемых: f_j^5; $\sum f_k f_l^4$, где $k + 4l = 5j$; $\sum f_k f_l f_m^3$ где $k + l + 3m = 5j$, $k \neq l \neq m \neq k$, но неверно, что $k \equiv l \equiv m \pmod 5$; $\sum f_k f_l^2 f_m^2$, где $k + 2l + 2m = 5j$, $k \neq l \neq m \neq k$, но неверно, что $k \equiv l \equiv m \pmod 5$; $\sum f_k^2 f_l f_m f_n$, где $2k + l + m + n = 5j$, $k \equiv l \not\equiv m \not\equiv n \not\equiv k \pmod 5$, причем $|\{k, l, m, n\}| = 4$; Наконец, $\sum f_k f_l f_m f_n f_i$, где $k + l + m + n + i = 5j$, и числа k, l, m, n, i различны и не сравнимы по модулю 5.

Как видно, эффективность этих формул в том, что коэффициент b_j вычисляется только через начальный отрезок ряда коэффициентов a_0, a_1, ..., a_{pj} длины pj (в чем и заключается рекуррентность).

Так, вычисляя f_p для $p = 3$, $p = 5$, получаем всевозможные БЧХ-коды, определяющие 6 ошибок и исправляющие 3 ошибки, с кодированием посредством полинома $F(x) = f(x) f_3(x) f_5(x)$, где $f(x)$ – произвольный примитивный многочлен. Если степень $f(x)$ равна восьми, получаем все возможные варианты кодирования, соответствующие европейскому стандарту передачи данных по сетям.

Этот алгоритм применим также к задаче перечисления всех неприводимых многочленов данной степени n при известном одном примитивном многочлене $f(x)$ в случае, например, когда p – первообразный по модулю $2^n - 1$. В этом случае можно также определить максимальную длину посылаемого помехоустойчивого кода (в битах) для данного кодового расстояния. Если $d_{min} = 3$, то, как известно, эта длина равна $2^n - 1$. Если $d_{min} > 3$, то можно выписать это число в терминах соответствующего логарифма Зеха-Якоби без перебора всей таблицы полиномиального кодирования.

Литература

[1] Яковлев В. В., Корниенко А. А. Информационная безопасность и защита информации в корпоративных сетях железнодорожного транспорта. Учебник для вузов ж.-д. транспорта / Под ред. В. В. Яковлева. М.: УМК МПС России. 2002. 328 с.

[2] Бабаш А. В., Шанкин Г. П. Криптография. Под редакцией И. П. Шерстюка, Э. А. Применко / Серия книг «Аспекты защиты». М.: СОЛОН-Р. 2002. 512 с.

[3] Шнайер Б. Прикладная криптография. 2-е издание: протоколы, алгоритмы и исходные тексты на языке С. 1996. (пер. с англ.: Bruce Schneier. Applied Cryptography. Second Edition: Protocols, Algorithms, and Source Codes in C. John Wiley & Sons. 1996. 758 p.)

[4] Лидл Р., Нидеррайтер Г. Конечные поля. М.: Мир. 1988.

[5] Масленников М. Е. Практическая криптография. СПб: БХВ-Петербург. 2003. 464 с. + CD.

[6] Демкина О. Е., Титов С. С., Торгашова А. В. Рекуррентное вычисление коэффициентов степеней экспоненты. Проблемы теоретической и прикладной математики. Труды 34-й Региональной молодежной конференции. Екатеринбург: УрО РАН. 2003. С. 27–30.

[7] Бухштаб А. А. Теория чисел. М.: Просвещение. 1966. 384 с.

Алгоритм расщепления спектра для проверки изоморфизма графов и его приложения

А. В. Пролубников, Р. Т. Файзуллин

Задача проверки изоморфизма графов принадлежит к задачам, относительно которых нет ясности: являются ли они полиномиально разрешимыми или нет [1]. Известно, что задача полиномиально разрешима для некоторых классов графов. В частности, для планарных, графов с ограниченной степенью вершин, графов с ограниченной кратностью собственных значений их матриц смежности и некоторых других построены эффективные алгоритмы решения задачи проверки изоморфизма графов [2], [3], [4].

Если графы изоморфны, то возможно получение матрицы смежности одного графа некоторой перестановкой строк с такой же перестановкой столбцов (перестановкой рядов) матрицы смежности второго графа, и допустима следующая постановка задачи проверки изоморфизма графов, эквивалентная приведенной выше: даны матрицы смежности графов — A_0 и B_0. Требуется найти матрицу перестановки P такую, что $A_0 = PB_0P^{-1}$, или показать, что такой матрицы перестановки не существует. Если спектры матриц смежности $\mathrm{Sp}(A_0)$ и $\mathrm{Sp}(A_0)$ просты, то задача построения изоморфизма становится тривиальной и реализуется перестановкой вершин соответствующей перестановке рядов матриц собственных векторов A_0 и B_0 [5]. Группа автоморфизмов графа (изоморфных отображение множества вершин графа на себя) реализуется множеством всех матриц перестановок, которые коммутируют с матрицей смежности графа. Если группа автоморфизмов $\Gamma(GA)$ графа не тривиальна, что соответствует наличию симметрий в графе относительно перестановок его вершин, то $\mathrm{Sp}(A_0)$ содержит кратные собственные значения [6], и приведенный выше критерий места не имеет: матрица P — не единственна, и установление изоморфизма покомпонентным сравнением собственных векторов невозможно. Возможно возмущение матриц смежности обоих графов до некоторых матриц A и B таких, что и $\mathrm{Sp}(B)$, $\mathrm{Sp}(A)$ будут простыми [5]. Однако, вычисление спектра и всех собственных векторов матрицы, а главное, вычисление матрицы, при помощи которой должно быть произведено возмущение, — задача вычислительно значительно более тяжелая, чем решение систем линейных алгебраических уравнений, с которыми работает рассматриваемый ниже алгоритм спектрального расщепления проверки изоморфизма графов.

Предлагаемый нами алгоритм работает с модифицированными матрицами смежности графов и основан на решении связанных с ними систем линейных алгебраических уравнений. Построение изоморфизма, если графы изоморфны, происходит на итерациях алгоритма без осуществления ветвления в соответствии с некоторым деревом поиска. Изоморфны графы или нет устанавливается не более чем за n итераций алгоритма, где n — число вершин в графах. Графам GA и GB ставятся в соответствие положительно определенные матрицы с диагональным преобладанием. На итерациях алгоритма решаются системы линейных уравнений, задающие матрицы A^{-1}, B^{-1}. Последовательно возмущая

диагонали матриц A и B в ходе работы алгоритма, мы разрушаем симметрии в графе и не более чем за n итераций приходим к ситуации, когда возможно установление изоморфизма графов. На итерациях алгоритма происходит последовательное возмущение рабочих матриц алгоритма, такое, что сохраняется возможность получения матриц, поставленных в соответствие графам, некоторой перестановкой их рядов. Возмущая матрицы, удается достичь численно эффективного расщепления как спектров матриц, так и расщепления норм решений систем линейных уравнений, что позволяет в случае изоморфизма графов установить взаимно однозначное соответствие. Расщепление достигается при заданной на старте алгоритма длине мантиссы машинных чисел и заданном числе итераций решения систем линейных уравнений. Доказана оценка границ интервала, в котором происходит расщепление множеств, относительно возмущений матриц, позволяющая определить трудоемкость алгоритма как полиномиальную для широкого класса графов, включая указанные в [2], [3], [4], составляющую в наиболее сложных случаях $O(n^5)$.

На основе эвристики, используемой для проверки изоморфизма графов, построен алгоритм поиска оптимального вложения графов — решения задачи, представляющей собой обобщение задачи поиска в данном графе подграфа, изоморфного другому графу. Модификация алгоритма применима к решению задач проверки изоморфизма ориентированных и взвешенных графов, что позволяет эффективно применять алгоритм к решению задачи дешифрования шифра двойной перестановки [7], являющимся обобщением одной из базовых процедур шифрования данных — шифра перестановки. Так в ходе численных экспериментов показано, что число символов в дешифруемом тексте может достигать 10000 при устойчивой работе алгоритма. Также рассматривается приложение построенного алгоритма дешифрования шифра двойной перестановки к следующей задаче. По каналу связи от источника к приемнику передается видеоизображение. Необходимо шифровать видеоизображение для сокрытия информации при несанкционированном подключении третьих лиц к каналу связи. При этом необходимо без потери эффективности процедуры дешифрования реализовать шифрование изображения так, чтобы ключ к шифру динамически менялся при передаче кадров видеоизображения по каналу связи, без передачи ключа к шифру от источника к приемнику в явном виде, и без знания текущего ключа источником. В целях повышения эффективности шифрования можно использовать дополнительную процедуру искажения шифрованных данных, состоящую в некотором возмущении цветовых характеристик передаваемых кадров видеоизображения, с также динамически изменяемыми параметрами возмущения. В ходе передачи видеоизображения эти параметры могут варьироваться в пределах заданных интервалов, что, внося искажения в передаваемые данные с последующим восстановлением их получателем, позволяет оставлять схему дешифрования неизменной на протяжении всего сеанса связи. При этом используется вычислительная устойчивость алгоритма спектрального расщепления проверки изоморфизма графов относительно этой процедуры при заданных границах на параметры возмущения.

Литература

[1] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.

[2] Hopcroft J., Wong J. A linear time algorithm for isomorphism of planar graphs. Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, 1974. P. 172–184.

[3] Luks E. M. Isomorphism of graphs of bounded valence can be tested in polynomial time. Proc. 21st IEEE FOCS Symp, 1980. P. 42–49.

[4] Hoffmann C.M. Group-Theoretic Algorithms and Graph Isomorphism Lecture Notes in Computer Science (Chapter V), 1982. P. 127–138.

[5] Кикина А. Ю., Файзуллин Р. Т. Алгоритм проверки изоморфности графов. Деп. ВИНИТИ 21.06.95, 1789-В95.

[6] Цветкович Д. и др. Спектры графов. Теория и применение. Киев: Наукова думка, 1984.

[7] Faizullin R., Prolubnikov A. An Algorithm of the Spectral Splitting for the Double Permutation Cipher. Pattern Recognition and Image Analysis. MAIK, Nauka. Vol. 12, p. 365–375. No. 4, 2002.

О треугольных преобразованиях специального вида[9]

М. В. Федюкин

Пусть $\Omega_{2^n} = \{0, 1, \dots 2^n - 1\}$. Для разбиения $\gamma = i_0, 1, \dots i_1 - 1\} \cup \{i_1, i_1 + 1, \dots, i_2 - 1 \cup \dots \cup \{i_{t-1}, i_{t-1} + 1, \dots, i_t - 1\}$, где $i_0 = 0 < i_1 < i_2 < \dots < i_t = n$, множества $\{0, 1, \dots, n-1\}$ определим бинарную операцию θ_γ на множестве Ω_{2^n} следующим образом. Положим, что для элементов $a = \sum_{i=0}^{n-1} a_i 2^i$, $b = \sum_{i=0}^{n-1} b_i 2^i$, $c = \sum_{i=0}^{n-1} c_i 2^i$, $a_i, b_i, c_i \in \mathrm{GF}(2)$ равенство $a\,\theta_\gamma\, b = c$ имеет место тогда и только тогда, когда для любого $0 \leqslant s \leqslant t-1$ выполнено равенство

$$\sum_{j=i_s}^{i_{s+1}} a_j 2^{j-s} + \sum_{j=i_s}^{i_{s+1}} b_j 2^{j-s} = \sum_{j=i_s}^{i_{s+1}} c_j 2^{j-s}$$

Обозначим через Γ_n множество всех разбиений множества Ω_{2^n} подобного вида. Операции из множества $\Theta_n = \{\theta_\gamma \mid \gamma \in \Gamma_n\}$ называются операциями модульного сложения. Они являются частным случаем биективных треугольных преобразований.

Назовем высотой $\alpha(x_{i,j})$ переменной $x_{i,j}$ число 2^j, $\alpha(0) = \alpha(1) = 0$. Высота $\alpha(x_{i,j}x_{k,l}\dots x_{s,t})$ монома $x_{i,j}x_{k,l}\dots x_{s,t}$ равна сумме высот переменных, входящих в него, а высота $\alpha(\psi)$ полинома ψ равна максимуму высот мономов этого полинома.

Рассмотрим произвольное разбиение δ множества $\{0, 1, \dots, n-1\}$ на два подмножества Δ_0 и Δ_1. Пусть всегда $0 \in \Delta_0$. Построим по этому разбиению бинарную операцию $*_\delta$ следующим образом. Положим для $a = \sum_{i=0}^{n-1} a_i 2^i$, $b = \sum_{i=0}^{n-1} b_i 2^i$, $c = \sum_{i=0}^{n-1} c_i 2^i$, $a_i, b_i, c_i \in \mathrm{GF}(2)$, что равенство $a *_\delta b = c$ имеет место тогда и только тогда, когда $c_i = a_i + b_i + a_{i-1}b_{i-1}$ при $i \in \Delta_1$ и $c_i = a_i + b_i$ при $i \in \Delta_0$. По разбиению γ из Γ_n однозначно строится разбиение $\Delta_0 \cup \Delta_1$ и обратно. Обозначим множество всех разбиений δ указанного вида через D_n.

Двоичной нормой $\|a\|_2$ элемента $a \in \Omega_{2^n}$ называется такое неотрицательное целое число t, что 2^t делит a, а 2^{t+1} не делит a.

Обозначим множество функций от переменных $x_1, x_2, \dots, x_m$ над алгеброй $\Omega_{2^n}(\{*_\delta \mid \delta \in D_n\})$ через $W_{2^n,m}(D_n)$.

Утверждение 1. *Функция $\varphi = (\varphi_0, \varphi_1, \dots, \varphi_{n-1})$ содержится в множестве $W_{2^n,m}(D_n)$ тогда и только тогда, когда*

$$\varphi_i = \sum_{j=1}^{m} a_j x_{j,i} \oplus \psi_i,$$

где $a_j \in \mathrm{GF}(2)$, $j = 1, 2, \dots m$, $\alpha(\psi_i) \leqslant 2^i$, $a\psi_i$ не зависит от $x_{j,i}$, $j = 1, 2, \dots, m$.

Поставим в соответствие каждой операции из множества $\{*_\delta \mid \delta \in D_n\}$ вектор из векторного пространства V_n следующим образом. Положим, что разбиению δ соответствует вектор $v_\delta = (v_0, v_1, \dots, v_{n-1})$, где $v_i = 1$ тогда и только тогда, когда $i \in \Delta_1$. Очевидно, что вектора, соответствующие всем операциям из множества $\{*_\delta \mid \delta \in D_n\}$, порождают векторное пространство размерности $n-1$ равное $\langle e_1, e_2, \dots, e_{n-1}\rangle$, где e_i — элемент стандартного базиса векторного пространства V_n, в котором i-ая компонента (с учетом начала нумерации с 0) равна 1.

Утверждение 2. *Пусть $F \subseteq D_n$. Множество функций от m переменных над универсальной алгеброй $\Omega_{2^n}(\{*_\delta \mid \delta \in D_n\})$ равно $W_{2^n,m}(D_n)$ тогда и только тогда, когда выполнено равенство*

$$\langle \{v_\delta \oplus v_\eta \mid \delta, \eta \in F\}\rangle = \langle e_1, e_2, \dots, e_{n-1}\rangle.$$

Аналогичные утверждения для множества операций модульного сложения были получены автором в 1988 году.

[9]Работа выполнена при поддержке гранта Президента России № НШ 2358.2003.09.

Литература

[1] Кон П. Универсальная алгебра. М.: Мир, 1968, 352 с.

[2] Lausch H., Nöbauer W. Algebra of polynomials. Amsterdam-London: North.-Holl. Publ., 1973.

О критериях отсутствия ограниченных гомоморфизмов n-квазигрупп

И. Г. Шапошников

В связи с возможностью использования гомоморфизмов многоосновных универсальных алгебр при анализе криптографических алгоритмов (см. [1]) в ряде работ было предложено развитие понятия гомоморфизма, например, π-гомоморфизм [2], скрещенный гомоморфизм [3], гомоморфное отношение [4]. Напомним определение гомоморфного отношения для многоосновных универсальных алгебр $(A_1, A_2, \ldots, A_n, A_0; \omega)$ с одной операцией $\omega\colon A_1 \times A_2 \times \ldots \times A_n \to A_0$.

Определение ([4]). Пусть для n-арных операций $\omega : A_1 \times A_2 \times \ldots \times A_n \to A_0$ и $\vartheta : B_1 \times B_2 \times \ldots \times B_n \to B_0$, $|A_i| \geqslant |B_i|$, определены отображения φ_i множеств A_i на B_i и задано бинарное отношение ρ на векторах множества B_0^N, $N = |A_1 \times \ldots \times A_n|$, такие, что вектора

$$\{\varphi_0(\omega(a_1, \ldots, a_n)) : (a_1, \ldots, a_n) \in A_1 \times \ldots \times A_n\} \quad \text{и}$$
$$\{\vartheta(\varphi_1(a_1), \ldots, \varphi_n(a_n)) : (a_1, \ldots, a_n) \in A_1 \times \ldots \times A_n\}$$

находятся в отношении ρ. Тогда будем говорить, что многоосновная универсальная алгебра $(A_1, A_2, \ldots, A_n, A_0; \omega)$ находится в гомоморфном отношении ρ с алгеброй $(B_1, B_2, \ldots, B_n, B_0; \vartheta)$ при гомоморфизме отношения ρ $(\varphi_1, \varphi_2, \ldots, \varphi_n, \varphi_0)$.

Гомоморфными отношениями являются гомоморфизмы и π-гомоморфизмы. В [4] приведены некоторые другие частные случаи гомоморфных отношений. Если ρ — отношение покоординатного неравенства, то такое гомоморфное отношение названо инверсным гомоморфизмом. Если соотношение гомоморфизма выполняется на подмножестве Q множества определения $A_1 \times \cdots \times A_n$, то такое гомоморфное отношение названо неточным гомоморфизмом, а величина $p = |Q|/|A_1 \times \ldots \times A_n|$ — его значимостью. Если при этом подмножество Q есть объединение подмножеств вида $\alpha_1 \times \cdots \times \alpha_n$, где α_i — класс эквивалентности относительно $\operatorname{Ker}\varphi_i$, то такое гомоморфное отношение названо ограниченным гомоморфизмом на ограничении Q.

Рассмотрим определенные в [5] конечные n-квазигруппы $(A; F)$, построенные с использованием суперпозиций квазигрупп, изотопных некоторой одной квазигруппе $(A; \cdot) : F(x_1 x_2 \ldots x_n) = ((\ldots((x_1 h_1) \cdot x_2) h_2) \ldots \cdot x_n) h_n$, $x_i \in A$, h_i — подстановка на множестве A, fh — суперпозиция вида $h(f)$. Данный класс n-квазигрупп появляется и в криптографии. В частности, структуру такой n-квазигруппы задает преобразование блочного шифра, построенного по принципу Square. В [6] рассматривались неточные и ограниченные гомоморфизмы данного класса n-квазигрупп, в представленной работе приводятся результаты дальнейших исследований. Пусть $(A; \cdot)$ — абелева группа, φ_i, $i = 1, \ldots, n$, — ее гомоморфизмы, такие, что $|\operatorname{Ker}\varphi_i| = k$, $|A| > k > 1$. Обозначим через $M_l = M_l(\operatorname{Ker}\varphi_{i_l}, \operatorname{Ker}\varphi_{i_{l+1}})$ матрицу с элементами

$$q_{tj} = k^{-1} \cdot |h_l([a_t]\operatorname{Ker}\varphi_{i_l}) \cap [a_j]\operatorname{Ker}\varphi_{i_{l+1}}|,$$

$[a]\operatorname{Ker}\varphi_i$ — класс эквивалентности относительно $\operatorname{Ker}\varphi_i$.

В [6] были приведены условия отсутствия ограниченных гомоморфизмов $\Phi = (\varphi_{i_1}, \ldots, \varphi_{i_n}, \varphi_{i_0})$ n-квазигрупп $(A; F)$:

Утверждение 1 ([6]). *Пусть среди матриц M_l найдется $|A|/k - 1$ вполне неразложимых. Тогда Φ не является ограниченным гомоморфизмом $(A; F)$.*

Оставался открытым вопрос о выполнении этих условий, то есть о возможности построения подстановок h_l, для которых соответствующие матрицы M_l вполне неразложимы. Алгоритм такого построения для ряда групп $(A;\cdot)$ (в частности, для группы сложений кольца вычетов по модулю 2^N) следует из следующих трех утверждений.

Утверждение 2. *Пусть ε_i, μ_i, ε_{i+1}, μ_{i+1} — конгруэнции на группе $(A;\cdot)$, такие, что $|A/\varepsilon_i| = |A/\varepsilon_{i+1}|$, $|A/\mu_i| = |A/\mu_{i+1}|$, $\mu_i \subset \varepsilon_i$, $\mu_{i+1} \subset \varepsilon_{i+1}$. Тогда, если матрица $M(\varepsilon_i, \varepsilon_{i+1})$, частично разложима, то матрица $M(\mu_i, \mu_{i+1})$ также частично разложима.*

Утверждение 3. *Пусть ε и μ — конгруэнции на группе $(A;\cdot)$, такие, что $|A/\varepsilon| = |A/\mu| = |A|/2$. Тогда для любой подстановки h матрица $M(\varepsilon, \mu)$ представима выпуклой комбинацией двух матриц перестановок P_1, P_2 вида:*

$$M(\varepsilon, \mu) = \frac{1}{2}P_1 + \frac{1}{2}P_2.$$

Утверждение 4. *Если квадратная матрица M размера $m \times m$ представима выпуклой комбинацией двух матриц перестановок P_1, P_2 вида:*

$$M = \frac{1}{2}P_1 + \frac{1}{2}P_2,$$

то она вполне неразложима тогда и только тогда, когда матрице перестановки $P^{-1}P_2$ соответствует подстановка полного цикла симметрической группы S_m.

Приведем еще один критерий отсутствия ограниченных гомоморфизмов Φ n-квазигрупп $(A;F)$ для случая, когда $A = V_N$ — множество N-разрядных двоичных чисел, $|A|/k = 2^r$, $0 < r < N$:

Утверждение 5. *Если среди матриц M_l найдется $s > \log_{1-2^{-N+r}} 2^{-r}$ с положительными коэффициентами эргодичности, то Φ не является ограниченным гомоморфизмом $(A;F)$.*

Наиболее же просто выглядит критерий отсутствия ограниченных гомоморфизмов Φ n-квазигрупп $(A;F)$, в случае, когда $A = V_N$, $\cdot$ — операция сложения по модулю 2^N, а подстановки h_i, $i = 1, \ldots, n$, совпадают и задаются некоторой перестановкой $\varkappa$ координат N-мерных двоичных векторов.

Утверждение 6. *При $n > N$ Φ не является ограниченным гомоморфизмом $(A;F)$ тогда и только тогда, когда $\varkappa$ — полный цикл.*

Литература

[1] Горчинский Ю. Н. О гомоморфизмах многоосновных универсальных алгебр в связи с криптографическими применениями. М.: ТВП, Труды по дискретной математике, 1997, том 1, с. 67–84.

[2] Горчинский Ю. Н. О π-гомоморфизмах конечных многоосновных универсальных алгебр. Дискретная математика, 1999, том 11, вып. 2, с. 3–19.

[3] Карпунин Г. А., Шапошников И. Г. Скрещенные гомоморфизмы конечных многоосновных универсальных алгебр с бинарными операциями. Дискретная математика, 2000, том 12, вып. 2, с. 66–84.

[4] Шапошников И. Г. Гомоморфные отношения многоосновных универсальных алгебр. Тезисы докладов на четвертом Всероссийском симпозиуме по прикладной и промышленной математике, 2003.

[5] Глухов М. М. Об α-замкнутых классах и α-полных системах функций k-значной логики. — Дискретная математика, 1989, том 1, вып. 1, с. 16–21.

[6] Шапошников И. Г. О некоторых гомоморфных отношениях n-квазигрупп. Тезисы докладов на четвертом Всероссийском симпозиуме по прикладной и промышленной математике, 2003.

Роль скрытых каналов при построении защиты в распределенных компьютерных системах[10]

А. А. Грушо, Е. Е. Тимонина

Канал называется скрытым, если он не проектировался, не предполагался для передачи информации в электронной системе обработки данных [8]. Разумеется, необходимо приложить определенные усилия, для того чтобы заложенные производителем возможности можно было использовать для организации связи между некоторыми абонентами внутри компьютерной системы или между компьютерными системами через сеть. Скрытый канал предполагает, что передача информации по нему происходит незаметно для тех субъектов, которые были бы заинтересованы в его выявлении. Мы будем считать, что все эти субъекты сосредоточены в средствах защиты и скрытый канал должен обладать тем свойством, что средства защиты его «не видят». Сокрытие самого факта передачи информации имеет длинную историю и часто относится к области, называемой стеганографией. Но, постольку мы говорим о внутри или меж компьютерном обмене скрытой информацией, в котором отправителями и получателями информации могут быть программно-аппаратные агенты, а не пользователи, то мы предпочитаем использовать термин, возникший в компьютерной среде - скрытые каналы. Для простоты мы будем предполагать в дальнейшем, отправителями и получателями информации являются программно-аппаратные агенты нарушителя безопасности, которые могут находиться как в различных процессорах, связанных между собой шинами, так и в различных компьютерах, связанных через сеть. Мы не будем рассматривать вопрос, как попали эти агенты в ту или иную среду (см., например, [3]). Существует расхожее мнение, что скрытые каналы легко выявляются или уничтожаются с помощью контроля за системами передачи информации. Однако это не так. Приведем некоторые примеры скрытых каналов, которые не устранимы в компьютерных системах или сетях.

В сетях с коммутацией пакетов Proxy-серверы могут шифровать пакеты, устанавливать единую длину пакетов, задавать новые адреса, которые являются адресами шлюзов соответствующих сегментов локальных сетей, видоизменять другие параметры полей пакетов. В то же время неустранимой является связь между адресами шлюзов и внутренними адресами сегментов, выходящих на эти шлюзы. Как показано в работах [4], [7] этой связи оказывается достаточно для того, чтобы преодолеть реализуемую Proxy-сервером защиту и организовать связь между программно-аппаратным агентом в защищенном сегменте локальной сети и программно-аппаратным агентом в глобальной сети (например, Интернет). Данный канал возникает из-за необходимости адресации пакетов и, следовательно, является неустранимым, даже при использовании шифрования и модификации параметров передаваемых пакетов.

Другой пример неустранимого канала представляет собой модуляцию статистических характеристик потоков пакетов. Хотя такие каналы обладают низкой пропускной способностью, они позволяют преодолевать большинство существующих систем защиты на стыках защищаемых локальных сетей и глобальных сетей.

Наличие скрытых каналов, позволяющих обмениваться информацией программно-аппаратным агентам нарушителя безопасности «невидимо» для средств защиты, создает ситуацию неконтролируемого несанкционированного доступа к информации. Причем в некоторых моделях можно доказать, что традиционные способы внесения защиты в готовые продукты в принципе не могут обнаружить как наличие программно-аппаратных агентов, так и наличие каналов связи между ними. В таких ситуациях мы приходим к парадоксальному выводу о том, что система защиты может соответствовать нормативной базе, но вовсе не защищать информацию от утечки или разрушения. Отсюда следует вывод, что для защиты наиболее ценной информации необходимо индивидуальное обоснование защищенности либо в виде доказательства отсутствия программно-аппаратных агентов противника, либо в виде обоснования того, что такие агенты не могут нанести ущерб ценной информации. Для одних систем такое обоснование допустимо проводить неформальными методами, для критических систем

[10]Работа поддержана грантом РФФИ № 01-01-00895.

такое обоснование должно строиться на строгих формальных моделях и математическом доказательстве защищенности. Примеры формального доказательства защищенности имеются в [1].

Если сравнивать возможности закладки производителем программно-аппаратных агентов в создаваемые ими компьютерные системы с нашими возможностями анализа скрытых каналов, то мы приходим к следующему выводу. Мы умеем выявлять все потенциально возможные скрытые каналы и эффективно бороться с неустранимыми каналами. Однако мы не умеем проводить анализ программно-аппаратных средств настолько, чтобы полностью исключить существование программно-аппаратных агентов в компьютерной среде (например, в процессоре). На основании этого предлагается изменить парадигму построения систем защиты, как предупреждение несанкционированного доступа к информации, особенно при реализации многоуровневой политики безопасности, определяющей взаимодействие объектов и субъектов с различными грифами секретности.

Из предыдущего следует необходимость построения новой концепции информационной безопасности информационных технологий, реализуемой в условиях ограниченной информации о процессорах и программном обеспечении и в то же время обеспечивающей основные принципы многоуровневой политики, позволяющей нейтрализовать действия программно-аппаратных агентов нарушителя безопасности. Концепция контроля каналов допускает неполную информацию о процессорах и программном обеспечении, а также наличие и функционирование в процессорах и программном обеспечении программно-аппаратных агентов нарушителя безопасности. Изложим положения концепции для многоуровневой политики. Мы предполагаем, что все объекты системы могут быть классифицированы по грифам секретности и разрешены только информационные потоки от менее секретных объектов к более секретным [1]. Определим понятие одноуровневой подсистемы информационной системы. В одноуровневой системе вся информация (включая программное обеспечение и аппаратную платформу) классифицирована одним грифом секретности, и все пользователи данной подсистемы имеют допуск к информации данного класса. Таким образом, любые информационные потоки в одноуровневой подсистеме являются разрешенными с точки зрения многоуровневой политики. При этом допускается ограничения, связанные с дискреционным управлением доступом, который, как известно, принципиально не может быть надежным. Поэтому дискреционные ограничения управления доступом мы далее не рассматриваем. Из разрешенности информационных потоков в одноуровневой подсистеме следует:

- программно-аппаратные агенты нарушителя безопасности, если они имеются в подсистеме, могут иметь доступ к любой информации и перезаписывать ее в любой объект, однако нарушения конфиденциальности при этом не происходит;
- в одноуровневой подсистеме нецелесообразно делать какие бы то ни было механизмы контроля управления информационными потоками.

В концепции контроля каналов мы предполагаем, что система представима в виде одноуровневых подсистем, соединенных между собой каналами связи. Основная аксиома концепции контроля каналов состоит в том, что если мы контролируем и управляем всеми каналами между одноуровневыми объектами (включая скрытые каналы), то система может быть сделана безопасной.

Определим понятие однонаправленного канала следующим образом. Пусть отправитель и получатель обладают следующими свойствами. Отправитель может послать любое сообщение получателю, однако получатель не может передавать отправителю какую-либо информацию, кроме известной отправителю, или считающейся разрешенной для данных двух абонентов. Мы считаем, что от получателя к отправителю нет никаких каналов (даже скрытых), позволяющих нарушать указанные правила.

Тогда, если одноуровневые системы соединены однонаправленными каналами в направлении разрешенных потоков, то сделанные допущения эквивалентны определению многоуровневой политики [1], и здесь мы получаем не аксиому, а доказательное утверждение.

Аналогичный вывод можно сделать относительно политики Biba защиты целостности [1], [5].

Одноуровневую систему не следует рассматривать только как выделенный компьютер. На самом деле мы получаем гибкий инструмент реализации гарантированной защиты, так как одноуровневой системой можно считать любую подсистему информационной системы, в которой мы не можем детально провести анализ информационных потоков, но для которой мы можем провести полный анализ, связанных с ней каналов. Таким образом, одноуровневой системой может быть отдельная часть программного обеспечения (в случае, если имеем исходные коды и другую документацию и способны провести анализ), так и изолированная информационная система в целом, которая обрабатывает очень секретную информацию и отрезана от окружающей среды надежной защитой (включая отсутствие

всех каналов утечки). Однако многоуровневая политика и политика Biba не исчерпывают множества политик безопасности, которые можно реализовать на основе концепции контроля каналов. Например, на основе концепции контроля каналов можно реализовывать гибридные политики [2].

Определим безопасный интерфейс двух одноуровневых систем как возможность обмена информацией между ними таким образом, чтобы не нарушались требования по безопасности в каждой из подсистем и в системе в целом. Контроль каналов предполагает реализацию в каналах связи между одноуровневыми системами безопасные интерфейсы. Идея внесения ограничений на передачу информации в каналы для поддержки модели невлияния впервые была рассмотрена в работе [6]. Однако в данной работе рассматривается более широкое понятие безопасного интерфейса, позволяющее соединять одноуровневые системы, удовлетворяющие более широкому классу ограничений, например, ограничения реального времени, ограничения надежности обработки данных, ограничения дисциплины информационных технологий, устойчивость к сбоям и т.д. Разработка безопасных интерфейсов является самостоятельной задачей и в данной работе не рассматривается.

Теперь мы можем определить класс безопасных систем, который охватывает концепция контроля каналов. Система называется безопасной, если ее можно разделить на одноуровневые подсистемы, соединенные между собой каналами связи, которые обеспечивают безопасный интерфейс между ними, и других каналов в системе нет.

Рассмотрим управление безопасностью с помощью контроля каналов. В большинстве эффективных информационных технологий, автоматизирующих множество рутинных процедур, значительная часть требований по безопасности не меняется во времени. Однако всегда необходимо допускать несколько процентов отклонений от заданных правил. Эти отклонения можно реализовать с помощью фиксированных каналов для обращений к полномочным пользователям за разрешениями о передаче информации, не удовлетворяющей требованиям по безопасности в данной одноуровневой системе. При этом полномочный пользователь несет ответственность за нарушение принятой политики безопасности. Однако существуют информационные технологии, в которых требования по безопасности в одноуровневых системах могут меняться во времени. Для обеспечения таких политик требуются дополнительные механизмы очищения ресурсов и контроля за изменением настроек безопасных интерфейсов, каналов и одноуровневых систем. В этих случаях настройки каналов и безопасных интерфейсов должны управляться из некоторого интеллектуального центра безопасности с помощью защищенных систем удаленного администрирования. Богатство множества настроек, позволяющих обеспечить безопасность системы, реализуется при проектировании системы. Таким образом, концепция управления каналами позволяет реализовать достаточно широкий класс безопасных систем, включая системы с динамически меняющимися требованиями по безопасности к одноуровневым системам.

Литература

[1] Грушо А. А., Тимонина Е. Е. Теоретические основы защиты информации, М.: Агенство «Яхтсмен», 1996 г.

[2] Грушо А. А., Тимонина Е. Е. Гибридные политики безопасности, Тезисы докладов конференции «Методы и технические средства обеспечения безопасности информации», С.-Петербург, 1996.

[3] Грушо А. А., Тимонина Е. Е. О нормативно-методической базе по поиску скрытых каналов, Безопасность информационных технологий. Материалы научно-технической конференции органов по аттестации, аккредитованных в Системе сертификации Государственной технической комиссии, и организаций-лицензиатов по Приволжскому Федеральному округу, Пенза, сентябрь, 2002.

[4] Грушо А. А., Тимонина Е. Е. Языки в скрытых каналах, Труды международной конференции «Информационные технологии в науке, образовании, телекоммуникации, бизнесе (весенняя сессия)», Украина, Крым, Ялта-Гурзуф, 19–29 мая 2003 г.

[5] Biba K. J. Integrity Considerations for Secure Computer Systems, The MlTRE Corp., Report No. MTR-3153 Revision 1, Electronic Systems Division, U. S. Air Force Systems Command, Technical Report ESD-TR-76-372, Bedford, Massachusetts, April 1977.

[6] Goguen J. A., Meseguer J. Security Policies and Security Models, Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, pp. 11–20, April 1982.

[7] Grusho A., Timonina E. Construction of the Covert Channels, Information Assurance in Computer Networks. Methods, Models, and Architectures for Network Security: International Workshop MMM-ACNS 2003 St. Petersburg, Russia, LNCS 2776, Springer, 2003, 428–431.

[8] Lampson B. W. A Note of the Confinement Problem // Communications of ACM, 16:10, pp. 613-615, October 1973.

Математические основы методики автоматического доказательства для оценки безопасности информационных систем

Д. П. Зегжда, М. О. Калинин

Аннотация

Рассмотрен подход к решению проблемы безопасности с целью получения оценки защищенности информационных систем.

Безопасность систем обработки информации нуждается в формальном доказательстве, которое может гарантировать, что заданные ограничения на доступ к информации не будут нарушены. Доказательство того, что поведение системы не приводит к несанкционированному доступу, основывается на том, что реализованная в ней модель контроля и управления доступом является безопасной относительно начальной конфигурации системы.

В любой модели безопасности можно выделить три компонента: состояние, правила контроля доступа, критерии безопасности (рис. 1).

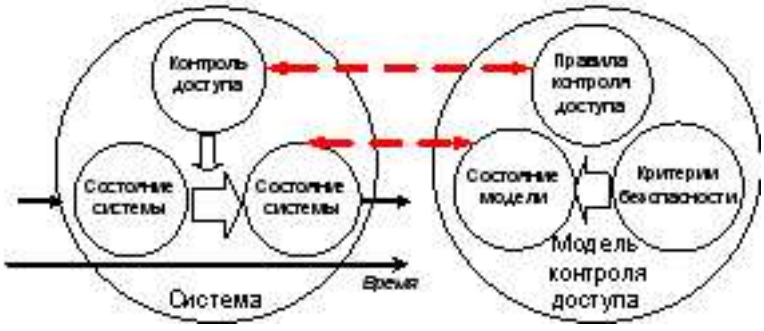


Рис. 1: Взаимосвязь компонентов системы и модели контроля доступа.

Описание состояниям системы, — это абстракция состояния системы в контексте модели, правила контроля доступа, определяют ограничения, накладываемые моделью на поведение системы, а критерии безопасности, позволяет выделить защищенные состояния из всего множества состояний.

Система считается безопасной в соответствии с моделью, если:

1) ее исходное состояние удовлетворяет критериям безопасности модели;

2) средства безопасности системы реализуют правила контроля доступа модели;

3) все состояния модели, достижимые из исходного, соответствуют критериям безопасности модели.

Процесс вычисления множества достижимых состояний и оценки их соответствия критериям безопасности называется *разрешением проблемы безопасности*. Разрешимость мандатных моделей доказана для общего случая. Для дискреционных моделей Харрисон, Руззо и Ульман показали, что проблема безопасности неразрешима для общего случая. В то же время для конкретных систем проведение такого доказательства возможно, и должно проводиться в ходе процесса оценки безопасности системы. Поскольку абсолютное большинство систем реализуют дискреционные модели, создание инструмента разрешения проблемы безопасности является актуальной задачей.

В самом общем виде проблема безопасности может быть формализована следующим образом:

Система Σ в общем виде представляет собой машину состояний: $\Sigma = \{S^\Sigma, T, s^\Sigma_{\text{init}}, Q\}$, где:

S^Σ — множество состояний системы;

Q — множество запросов, обрабатываемых системой;

T — функция перехода из состояний в состояние, T: $Q \times S^\Sigma \to S^\Sigma$. Функция T в ответ на запрос q переводит систему из состояния s^Σ_i в следующее $s^\Sigma_{i+1} = T(q, s^\Sigma)$;

s^Σ_{init} — начальное состояние системы.

Состояние s^Σ достижимо в системе $\Sigma = \{S^\Sigma, T, s^\Sigma_{\text{init}}, Q\}$ тогда и только тогда, когда существует последовательность $\langle (q_0, s^\Sigma_0), \ldots, (q_n, s^\Sigma_n) \rangle$, в которой $s^\Sigma_0 = s^\Sigma_{\text{init}}$, $s^\Sigma_n = s^\Sigma$, а $s^\Sigma_{i+1} = T(q_i, s^\Sigma_i)$, $0 \leqslant i < n$.

Модель безопасности M — это кортеж множеств: $M = \{S, R, C\}$, где:

S — множество состояний для данной модели;

R — множество правил контроля доступа, сформулированных в форме логических предикатов, определенных на множестве S, вида $r(s_1, s_2)$, определяющих допустимость перехода из состояния s_1 в состояние s_2 в соответствии с правилами модели;

C — множество критериев безопасности, сформулированных в форме логических предикатов вида $c(s)$, определяющих безопасность состояния s с точки зрения модели.

Состояние $s \in S$ является безопасным тогда и только тогда, когда для него истинны все критерии $c(s) \in C$, т. е. $\forall\, c \in C : c(s) =$ «истина».

Проблема безопасности представляется как $\Lambda = \{M, \Sigma, D\}$, где:

M — модель безопасности, $M = \{S, R, C\}$;

Σ — система, $\Sigma = \{S^\Sigma, T, s^\Sigma_{\text{init}}, Q\}$;

D — функция соответствия, D: $S^\Sigma \to S$, определяет соответствие между системными состояниями и состояниями модели.

На основе предложенных определений сформулирована *обобщенная теорема безопасности* систем обработки информации:

Система Σ, реализующая модель безопасности M, является безопасной тогда и только тогда, когда выполняются следующие условия:

1) для $\forall\, c \in C : c(D(s^\Sigma_{\text{init}})) =$ «истина»;

2) для $\forall\, s^\Sigma_i, s^\Sigma_{i+1} \in S^\Sigma : s^\Sigma_{i+1} = T(q, s^\Sigma_i)$ $\exists\, s_i, s_{i+1}$ такие, что $s_i = D(s^\Sigma_i)$, $s_{i+1} = D(s^\Sigma_{i+1})$ и для $\forall\, r \in R\ r(s_i, s_{i+1}) =$ «истина»;

3) для $\forall\, s^\Sigma_i \in S^\Sigma$, достижимого из состояния s^Σ_{init}, $\exists\, s_i$ такое, что $s_i = D(s^\Sigma_i)$ и для $\forall\, c \in C\ c(s_i) =$ «истина».

Следствиями данной теоремы являются постановки трех задач, которые могут быть решены методом разрешения проблемы безопасности:

1. *Проблема доказательства модели безопасности.* Для данной модели M требуется доказать, что любое достижимое состояние из множества состояний S соответствует критериям C, т. е. система на основе данной модели будет защищена в общем случае.

2. *Проблема построения защищенной системы.* Для данной системы Σ, критериев C, безопасных состояний S требуется построить множество правил контроля доступа R таких, что система Σ, реализующая модель контроля доступа $M = \{S, R, C\}$ была бы безопасной в соответствии с обобщенной теоремой безопасности.

3. *Проблема оценки защищенности системы.* Для данной системы Σ в состоянииs^Σ_{init}, использующей модель безопасности M, требуется произвести оценку безопасности всех достижимых состояний.

Рис. 2: Входные потоки решателя проблемы безопасности.

В докладе предлагается метод автоматизации решения проблемы оценки защищенности систем. Предложенный метод реализован в виде логического языка описания проблемы безопасности и специального инструментария анализа безопасности.

Язык описания проблемы безопасности (ЯОПБ) — это средство задания правил контроля доступа, критериев безопасности и состояний модели в форме логических предикатов с использованием синтаксиса языка Пролог. Выразительная способность языка позволяет описывать широкий класс моделей безопасности. Имеется опыт практического применения ЯОПБ для описания широко распространенных моделей безопасности, таких как модели Белла-ЛаПадулы, Харрисона-Руззо-Ульмана, ролевой контроль доступа и т. д.

Системные состояния и поведение системы представляются в виде *модель-ориентированного описания системных состояний* (МСС-описание). Правила контроля доступа, заданные с помощью ЯОПБ, формируют *описание правил контроля доступа* (ПКД-описание). Критерии безопасности, представленные на языке, составляют *описание критериев безопасности состояния* (КБС-описание).

Разработан специальный программный инструмент — «Решатель проблемы безопасности» (РПБ), представляющий собой машину логического вывода, построенную на основе Пролог-ядра, на вход которой подаются описанные на ЯОПБ состояние системы, правила контроля доступа и критерии безопасности (рис. 2).

Реализация РПБ позволяет промоделировать поведение системы, подчиняющейся заданной политике безопасности и оценить безопасность ее состояний. Разработаны программные средства, которые на основании описания политики безопасности, сформулированного с помощью предложенного языка, создают модель системы, подчиняющийся правилам этой политики, позволяют исследователю интерактивно изучать поведение системы в определенных ситуациях (попытки осуществления доступа, создание и уничтожение субъектов и объектов, изменение их атрибутов) и оценивать его корректность. Кроме того, такое исследование может выявить сильные и слабые стороны исследуемой политики безопасности.

Для автоматизации процесса оценки разработаны автоматизированные средства задания состояния системы и проведения тестирования защищенности (рис. 3): (1) анализатор системных состояний; (2) МСС-описание; (3) ПКД-описание; (4) КБС-описание; (5) менеджер критериев безопасности; (6) монитор незащищенных состояний; (7) генератор оценочных отчетов.

Оцениваемая система (например, операционная система) и модель безопасности, реализованная в ней, описываются на ЯОПБ в виде МСС-, ПКД-и КБС-описаний. Анализатор состояния проводит исследование системы и генерирует описание в соответствии с моделью контроля доступа. Если РПБ достигает состояния, в котором безопасность системы нарушена (состояние не соответствует критериям), то монитор незащищенных состояний позволяет показать последовательность событий, которые ведут к нарушению безопасности. Генератор отчетов строит отчет, содержащий описание

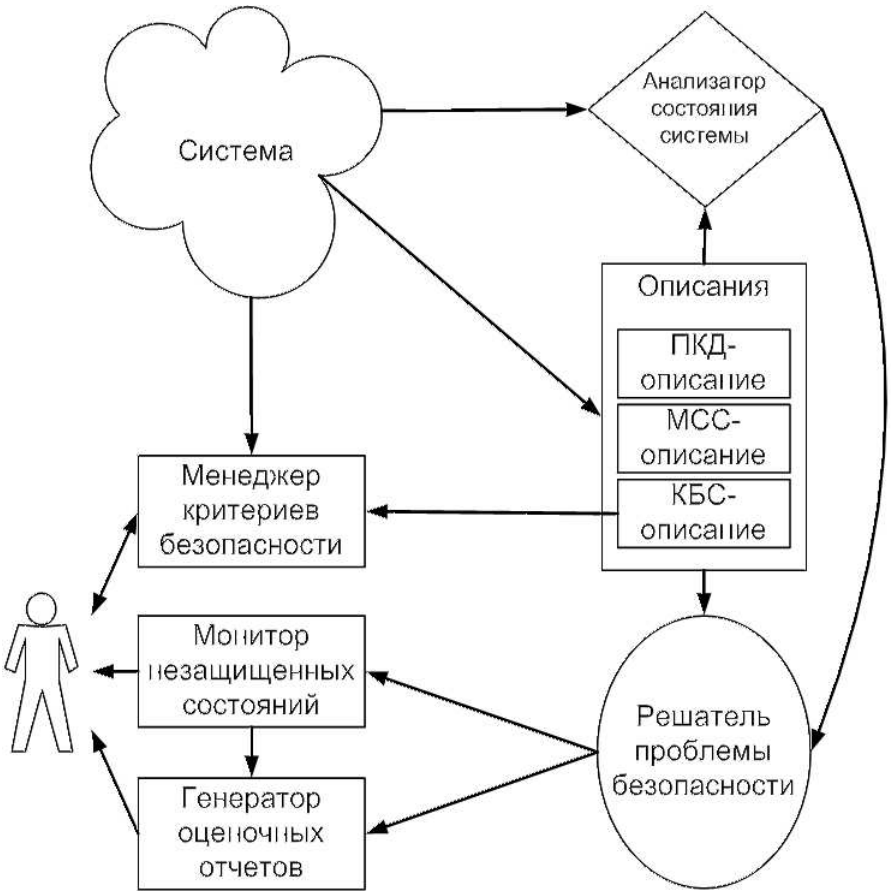


Рис. 3: Интегрированная система решения проблемы безопасности.

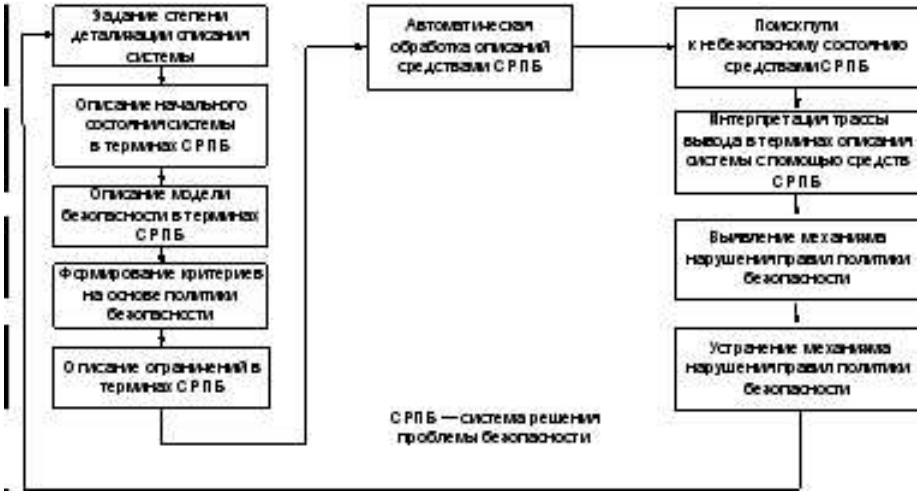


Рис. 4: Методика проверки выполнения правил политик безопасности.

модели, системы, начальное состояние, правила контроля доступа, критерии безопасности, результат оценки защищенности, трассу незащищенного состояния.

Основанный на предлагаемом подходе автоматический инструментарий дает возможность оценивать безопасность систем, механизмы контроля доступа которых базируются на реализации дискреционных моделей контроля: ОС MS Windows, ОС Linux, WWW-сервисы, межсетевые экраны.

Основу использования разработанного решения составляет методика применения системы решения проблемы безопасности (рис. 4).

На этапе предварительного анализа выполняется определение степени детализации системы. Для этого определяется, какие субъекты, объекты, атрибуты безопасности следует вносить в описание системных состояний. Здесь же определяется, какие правила контроля и управления доступом из реализованной в системе модели безопасности следует представить в форме предикатов и какие правила политики безопасности будут проверяться.

После определения уровня детализации создаются описания начального состояния и требований модели безопасности, формируются критерии и проводится их описание терминах системы проверки. Начальное состояние записывается в виде предикатов автоматически средствами анализатора системных состояний из состава системы проверки. Описание ограничений формируется из правил политики безопасности и задается экспертом посредством менеджера критериев. Описание правил модели безопасности, реализованной в системе, производится экспертом.

После того, как получены все логические описания, выполняется автоматический этап обработки описаний средствами системы проверки. Описания проходят обработку в РПБ, который интерпретирует их и производит анализ выполнения правил политики безопасности. Результатом проверки является итоговый отчет о выполнении этих правил, содержащий результаты проверки.

Процесс проверки считается успешно завершенным, если выполнены все правила политики безопасности. В случае нарушения какого-либо ограничения указывается состояние системы, в котором оно не выполнилось. При этом с помощью трассы логического вывода определяется механизм невыполнения политики безопасности. Генератор отчета из состава системы проверки позволяет выполнить этот поиск автоматически. Описания корректируются и вновь обрабатываются в системе проверки, чтобы повторным анализом подтвердить решение проблемы. Если невыполнение правила политики безопасности повторно не наблюдается, то причина его возникновения установлена верно и подтверждена повторной проверкой.

Предложенная методика позволила провести автоматизированный анализ выполнения политики безопасности в ОС Windows 2000/XP. Для данной системы были построены описания начального состояния, правил модели безопасности ОС Windows 2000/XP и критериев безопасности.

Описание начального состояния (МСС-описание) генерируется с помощью анализатора состояния системы (АСС), который является системно-зависимым приложением, поскольку ориентирован на анализ компонентов конкретной системы. Состояние любой ОС представляет собой совокупность сущностей (активных, называемых субъектами, и пассивных, называемых объектами) и их атрибу-

тов безопасности (прав доступа, разрешений и т. д.). Начальное состояние локальной ОС Windows 2000 представляет собой совокупность субъектов (учетных записей пользователей и групп), объектов (элементов файловой системы: файлов, каталогов, разделов, и элементов реестра: хивов, ключей) и атрибутов (прав доступа субъектов к объектам, т. е. записей DACL, членства в группах). Сбор информации об этих элементах ОС проводится в автоматическом режиме.

Логическая модель подсистемы КУД в ОС Microsoft Windows 2000 (ПКД-описание) — это представление алгоритма работы монитора обращений с использованием предикатов ЯОПБ. Все правила, заданные в модели, можно разбить на группы:

- проверка типа субъекта или объекта. Например, правило isFile('c:\\windows\\win.ini') выясняет существование в системе файла с именем 'c:\windows\win.ini';
- определение атрибутов безопасности субъектов и объектов, в т. ч. определение наличия прав доступа. Например, правило pListContents('Alice', 'c:\\windows') выясняет, разрешено ли пользователю с именем Alice чтение каталога 'c:\windows';
- определение возможности выполнения субъектом некоторой операции. Например, правило canDeleteFile('Alice', 'c:\\windows\\win.ini') выясняет, может ли в текущем состоянии системы пользователь Alice удалить файл 'c:\windows\win.ini';
- моделирование некоторого действия, переводящего систему в следующее состояние. Например, правило aDeleteFile('Alice', 'c:\\windows\\win.ini') удаляет файл 'c:\windows\win.ini', если пользователь Alice может в текущем состоянии удалить этот файл.

Описание критериев безопасности (КБС-описание) задается на основании реализуемой в системе политики безопасности. С точки зрения пользователя системы безопасное состояние системы — это состояние, удовлетворяющее политике безопасности. Правило политики безопасности — формализованное описание небезопасного состояния системы, логическая функция $F(S, O, SA)$, заданная на множествах субъектов S, объектов O и атрибутов безопасности SA. Функция F — комбинация простых логических высказываний, каждое из которых назовем критерием безопасности. Для интерактивного задания критерия необходимо установить параметры функции F и логическую зависимость между ними. Менеджер критериев решает данную задачу, позволяя задавать критерии в терминах состояния системы с использованием логических операций и квантификаторов. Для формализации применен ЯОПБ.

На вход менеджера критериев поступает «срез» текущего состояния системы безопасности, в котором содержится информация о субъектах, объектах системы и их атрибутах. На выходе менеджера создается файл с логически заданными критериями безопасности вида:

criterion_name(par_1,par_2,...,par_n):-

predicate_1 op

predicate_2 op

...

predicate_m.,

где criterion_name — имя (идентификатор) критерия, par_i — параметр критерия, predicate_i — предикат, составляющий правило критерия, op — логическая операция. В качестве параметра критерия может выступать переменная, что обеспечивает правила квантификации в соответствии с соглашениями Пролога.

Созданные описания обрабатываются с помощью средств системы решения проблемы безопасности, что позволило автоматически обнаружить состояния, не соответствующие политике, т. е. выявить небезопасные состояния исследуемой системы. Приведем примеры критериев: пользователь из группы Power Users может оперировать членством в группах для пользователей, которых сам не создавал; пользователь не может удалить каталог, которым владеет; пользователь, не имеющий членства в группах Administrators или Power Users, может создавать или модифицировать файлы в системном каталоге.

Для понимания нарушений политики безопасности или ошибок администрирования необходимо проследить последовательность переходов для логических предикатов критериев. Эту задачу решает

анализатор трассы логического вывода, использующий лингвистический подход, который заключается в фильтрации части трассы, значимой для объяснения причин нарушения безопасности. В трассе производится поиск лингвистических шаблонов, имеющих вид:

[predicate] [par_1_descr]... [par_n_descr],

где [predicate] — функтор; [par_1_descr] — словосочетание, которое будет использовано в отчете для представления параметра 1 предиката predicate; [par_n_descr] — словосочетание, которое будет использовано в отчете для представления параметра n предиката predicate. Например, шаблон [hasPermission][User][is authorized to][to the object] соответствует предикату hasPermission(administrator,write,sec_folder) и означает, что субъект administrator имеет право записи в каталог sec_folder. Данный предикат будет оттранслирован следующим образом:

User administrator is authorized to write to the object sec_folder.

Таким образом, результат анализатора трассы логического вывода — это языковые конструкции на понятном пользователю языке, абстрагирующие детали логического вывода о безопасности системы.

Разработанная авторами система решения проблемы безопасности может быть использована для решения следующих задач:

1. Оценка заданного состояния системы на соответствие заданным критериям безопасности. Входными параметрами являются описание состояния (МСС) и критерии (КБС), а выходом — оценка защищенности состояния.

2. Построение множества состояний анализируемой системы, достижимых из заданного, и оценка их безопасности. Входными параметрами будут описание текущего состояния (МСС), правила контроля доступа (ПКД) и критерии безопасности (КБС), на выходе получаются достижимые безопасные состояния.

Вероятностные модели гарантированно защищенных систем[11]

А. В. Галатенко

Для многопользовательских распределенных систем обычные детерминированные модели гарантированно защищенных систем оказываются слишком громоздкими — в них очень много состояний. Поэтому возникает необходимость уменьшения числа состояний за счет огрубления модели. Рассмотрим обобщение автоматной модели невлияния [1] на случай вероятностных автоматов и модель, ориентированную на обеспечение высокой пропускной способности каналов.

1 Обобщение модели невлияния на вероятностные автоматы

Под вероятностным автоматом понимается тройка $\mathbb{A} = (S, \Sigma, \delta)$, где S и Σ суть конечные множества (состояния и входной алфавит, соответственно), а δ — функция, определенная на множестве $S \times \Sigma$ и принимающая в качестве значений вероятностные меры на множестве S (обозначим множество таких мер через μ) ([2, 3]).

Пусть $|S| = n$. Тогда $\mu = \{(p_1 \dots p_n) \,|\, p_i \geqslant 0, i = 1 \dots n, \sum_{i=1}^{n} p_i = 1\}$. Функция δ может быть определена как некоторое множество стохастических матриц. Пусть $|\Sigma| = m$. Тогда $\delta = (M_1 \dots M_m)$.

[11]Работа частично выполнялась по программе фундаментальных научных исследований ОИТВС РАН «Математические модели, алгоритмы и инструментальные средства защиты ресурсов распределенных информационно-вычислительных систем».

Пусть в системе работают два пользователя: H и L. Пользователь H имеет право знать о системе все, а L — только часть информации. Пусть $\Sigma = \Sigma_H \sqcup \Sigma_L$. Пусть состояние — вектор параметров, часть которых соответствует пользователю L, а оставшаяся часть — пользователю H. Исходя из описанной идеологии, формально определим понятие безопасности.

Будем считать, что изначально система находится в безопасном состоянии.

Введем на S отношение эквивалентности, полагая $s_i \sim s_j$, если они отличаются только на H-компонентах. Соответствующие классы эквивалентности индуцируют проектор $\pi\colon S \to S/\sim$. Пусть $[s]$ — класс эквивалентности s. Введем функцию $F : \Sigma^* \to \Sigma_L^*$, определяемую так: если $\omega = x_1 \dots x_N$, то $F(\omega) = F(x_1) \dots F(x_N), F(x_i) = x_i$ при $x_i \in \Sigma_L$, и $F(x_i) = \varepsilon$ при $x_i \in \Sigma_H$.

Занумеруем состояния автомата $\mathbb{A}$ так, что сначала берутся состояния, соответствующие первому классу эквивалентности, затем — второму, и так далее. Тогда матрицы вида M_k могут считаться состоящими из блоков, соответствующих переходу из одного класса эквивалентности в другой. Пусть $|S/\sim| = l$. Обозначим через μ_L множество вероятностных мер на $S/\sim$. Рассмотрим проектор $\tau : \mu \to \mu_L$, задаваемый так. Если p_{i_j} соответствует состоянию из i-того класса эквивалентности, то полагаем $p_i^L = \sum_{j=1}^{k_i} p_{i_j}$, и при $\widetilde{\mu} \in \mu$ выполнено $\tau(\widetilde{\mu}) = (p_1^L \dots p_l^L)$.

Пусть $M(l)$ — множество квадратных матриц размеров $l \times l$. Определим функцию $Agr : \Sigma_L^* \to M(l)$ следующим образом. Пусть $\omega \in \Sigma_L^*$, $\omega = \omega(1) \dots \omega(k)$ и $M = M_{\omega(1)} \cdot \dots \cdot M_{\omega(k)}$.

В матрице M происходит объединение состояний, соответствующих одному классу эквивалентности так, что для каждого класса выбирается один представитель, и в качестве вероятности перехода из данного класса в другие берется суммарная вероятность попадания из выбранного состояния в состояния другого класса. Таким образом, от матрицы M размеров $n \times n$ переходим к матрице N размеров $l \times l$ и говорим, что $Agr(\omega) = N$.

Определение безопасности системы. Система, задаваемая вероятностным автоматом $\mathbb{A} = (S, \Sigma, \delta)$, называется безопасной, если выполнены следующие два условия:

1) отображение $\widehat{\delta}$, такое что $\widehat{\delta}(\widetilde{\mu}, \omega_L) = \tau(\widetilde{\mu}) \cdot Arg(\omega_L)$, корректно определено и действует в μ_L;

2) следующая диаграмма коммутативна:

$$\begin{array}{ccc} S \times \Sigma^* & \xrightarrow{\widetilde{\delta}} & \mu \\ \downarrow F & & \downarrow \tau \\ S \times \Sigma_L^* & \xrightarrow{\widehat{\delta}} & \mu_L \end{array}$$

ЯСформулируем достаточные условия безопасности системы. Будем говорить, что матрица вида M_k обладает свойством стационарности, если сумма элементов каждого блока по строке постоянна независимо от выбора строки в блоке. Матрица вида M_k обладает свойством диагональности, если все ее внедиагональные блоки состоят из нулей.

Теорема 1. *Для выполнения условия 1) безопасности системы достаточно, чтобы матрицы M_k, соответствующие каждому элементу Σ_L, были стационарными.*

Теорема 2. *Для выполнения условия 2) безопасности системы достаточно, чтобы матрицы M_k, соответствующие каждому элементу Σ_H, обладали свойством диагональности.*

2 Оценка интегральной загруженности каналов

Рассмотрим случайный граф. Пусть $\{x_1, \dots, x_n\}$ — множество вершин; ребра в графе появляются независимо, с одинаковой вероятностью $p, 0 \leqslant p\ le1$; в графе нет параллельных ребер и петель. Каждое значение p индуцирует вероятностную меру на графе; обозначим эту меру через P_p.

Пусть $N = C_n^2$, $0 \leqslant a \leqslant b \leqslant 1$, $A = aN$, $B = bN$. A и B соответствуют нижней и верхней границе допустимого числа ребер (естественно считать, что «нормальное» число ребер является долей общего числа ребер). Обозначим через V число ребер случайного графа.

Пусть $\{x_{i_1}, \dots, x_{i_k}\}$ — некоторое выделенное подмножество множества вершин. Без ограничения общности можно считать, что выделенные вершины имеют номера от 1 до k. Мы наблюдаем подграф, натянутый на $x_1, \dots, x_k$. Пусть ν — число ребер этого подграфа. По наблюдению ν требуется определить, лежит ли число ребер всего графа в описанных выше пределах.

Пусть $K = C_k^2, c_1 = aK, c_2 = bK$. Найдем, при каких k при стремлении n к бесконечности можно построить состоятельную оценку, а при каких — нет.

Теорема 3. *Пусть* $\frac{k}{\sqrt{n}} \to \infty (n \to \infty)$. *Тогда для любого* $\varepsilon, 0 < \varepsilon < 1$, *следующие события сходятся к* 0 *при* $n \to \infty$ *по вероятности в любой мере* $P_p, 0 \leqslant p \leqslant 1$:

$$(V > A, \nu < (1-\varepsilon)c_1), \tag{1}$$
$$(V < A, \nu < (1+\varepsilon)c_1), \tag{2}$$
$$(V > B, \nu < (1-\varepsilon)c_2), \tag{3}$$
$$(V < B, \nu < (1+\varepsilon)c_2). \tag{4}$$

Теорема 4. *Пусть* $k = o(\sqrt{n})(n \to \infty)$. *Тогда по наблюдению за числом ребер подграфа нельзя построить состоятельную оценку числа ребер всего графа.*

Литература

[1] Moskowitz Ira S., Costich Oliver L. A Classical Automata Approach to Noninterference Type Problems. Department of the Navy, Naval Research Laboraory, 1992.

[2] Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. М.: Наука, 1985.

[3] Бухараев Р. Г. Вероятностные автоматы. Казань, 1970.

Концепция программно-аппаратного комплекса «Тест»

Н. В. Макаров-Землянский, Б. В. Добров

1 Введение

Посредством Интернет происходит увеличение взаимодействия между людьми, что в Интернет равно обмену данными между компьютерами. При этом имеет место существенный рост сложности программного обеспечения, проявляющийся в росте функциональности, удобства использования, в конечном счете — в большом числе возможных реакций ПО на действия пользователя, состояние операционной системы, других программ и т. п. В силу разных причин в последние годы увеличивается использование недостаточно сертифицированных программ, так как пользователи отдают предпочтению удобству, не обращая внимание на реальные и потенциальные угрозы.

Те или иные данные, циркулирующие в сети, могут содержать в себе «команды», либо приводящие в действие вредоносное обеспечение (возможно даже без ведома отправителя), любо «просто» выводящее из строя компьютерную систему адресата из-за нового/неизвестного возникающего состояния в процессе обработки сообщения.

Поэтому в последнее время значительно возрос интерес к проблемам тестирования программного обеспечения. Методы тестирования [1] многообразны и зависят от задачи. Традиционно выделяют тестирование «белого ящика» (при наличии для анализа исходных текстов ПО) и тестирование «черного ящика» (только исполняемый код). При известном исходном коде выделяют статическое тестирование (только по коду программы) и динамическое тестирование (прохождение ветвей при выполнении программы для реальных наборов входных данных).

Проблема состоит в том, что в настоящее время сложность кода создаваемого программного обеспечения настолько велика — использование фрагментов кода сторонних производителей, быстрая

смена версий собственно программного продукта (при наличии недостаточно подробно оттестированных компонент), быстрая смена версий операционных систем (изменение окружения выполнения программы) — что исследование функциональности достаточно большой программы даже при наличие исходных кодов представляет собой серьезную задачу.

Данная работа описывает основные принципы, положенные в основу программно-аппаратного комплекса «Тест», разрабатываемого в НИВЦ МГУ и предназначенного для определения нежелательной функциональности сложных программных систем.

2 Основные требования на систему тестирования сложного программного обеспечения

Рассматривается задача тестирования программных комплексов [3], состоящих из большого числа компонент, имеющих развитую функциональность, что для исходных кодов означает разработку в течение долгого времени большим коллективом людей, то есть разнообразие стилей программирования, разнообразные программные решения, возможное дублирование функций и т. п. Требуется разобраться в функциональности программного обеспечения — определить наличие недокументированных возможностей, прежде всего «закладок», «люков», то есть фрагментов кода, приводящих к недокументированным действиям исследуемой программы, включая изменение логики работы программы, вредоносные функции. С другой стороны, необходимо проанализировать текст программы на наличие уязвимостей, потенциально способных вызывать сбой в работе ответственного программного обеспечения.

Программное обеспечение представляет собой: собственно исполняемый код (порождаемый в среде конкретных компиляторов и исполняемый в конкретной операционной системе), текст программы в исходных кодах, а также логику использования программы (фиксируемую в технических спецификациях различных уровней стандартизации). В результате для поддержки исследования сложного программного обеспечения необходимо поддерживать работу коллектива экспертов, которые имеют различную специализацию и квалификацию.

В последнее время большее развитие получают продукты, поддерживающие описание функциональности программного обеспечения с использованием универсальных языков описания функциональности UML, SDL. Лидером здесь является фирма Rational Rose, поставляющая на рынок линейку продуктов, позволяющих сопровождать полный цикл разработки ПО (планирование, разработку, тестирование). Однако, продукты Rational Rose предназначены прежде всего для поддержки процесса разработки ПО, кроме того ориентированы на среду MS Windows.

Комплекс «Тест», разрабатываемый в НИВЦ МГУ, ориентирован на востановление функциональности уже существующего ПО, прежде всего в сетевых средах, включающих программные комплексы, компоненты которых функционируют в разных операционных системах, в том числе, при недостаточном уровне покрытия документацией.

3 Решения, реализуемые в комплексе «Тест»

Программное обеспечение комплекса «Тест» имеет следующие составные части — собственно средства анализа исходного кода, средства описания требуемой функциональности, средства описания реальной функциональности, средства документирования процесса тестирования.

3.1 Аппарат исследования кода

Функции анализа исследуемого кода достаточно традиционны и включают в себя поддержку статического и динамического тестирования, а также анализ изменений при смене версий тестируемых программ. Основой анализа является «внутреннее представление» исходного кода — совокупность таблиц функций, переменных, результаты разбора команд, операций, арифметических операторов и т. п. Поддерживается анализ ПО, разработанного с использованием языков C и C++. При этом имеется возможность подключения внешних «компиляторов» для других языков. В настоящее время разрабатывается анализатор языка Pascal.

Для визуализации исходного кода имеются средства просмотра как текста кода программного обеспечения, включая ассоциированную информацию, так и графическое представление (блок-схемы) функций. Компонент «исследователь кода» отвечает за поддержку статического тестирования — отслеживание возможных маршрутов исполнения программы, определение «критических маршрутов», на которых выполняются условия специфицируемые экспертом-исследователем. Компонент «зонды» поддерживает динамическое тестирование — прохождение маршрутов исполнения программы для реальных наборов данных. «Верификатор» определяет различия между версиями исследуемого программного обеспечения.

3.2 Описание требуемой функциональности

Различается два вида требуемой функциональности — фиксируемая в технических спецификациях (техническом задании, руководствах пользователя, справочных материалах), а также функциональность, которая обычно не оговаривается, но подразумевается — отсутствие ошибок, отсутствие «закладок» и т. п. Для фиксации заявленной функциональности, содержащейся в технических спецификациях, используется аппарат «требований документации», по сути, фрагментов текста спецификаций, которые выделяются экспертом. Для задания иных видов требований на тестируемое ПО используется «план тестирования», пункты которого должны быть отработаны экспертами.

3.3 Описание реальной функциональности, документирование процесса тестирования

Результатом тестирования, в конечном счете, является некий «сертификат», в котором описываются условия и возможность применения исследуемого программного обеспечения. Сертификат может быть выдан и без всякого тестирования, на основе каких-то иных соображений (мнений, оценок, специальной информации), однако, для ответственных приложений необходимо иметь четкую картину функциональности используемого ПО. В связи с этим документирование тестирования, итоговый отчет, на основании которого вырабатывается сертификат, и является формальной целью тестирования.

В комплексе «Тест» для документирования процесса тестирования используется аппарат «сообщений экспертов». Сообщение эксперта — это описание в текстовом виде проблемы (или, наоборот, отсутствие проблемы), которую встретил эксперт, изучая программный код. Сообщения экспертов об одной проблеме группируются в «карточку эксперта». При создании нового сообщения (для выделяемого интерактивно фрагмента исходного кода) отслеживается пересечение с ранее созданными карточками.

Предусмотрены средства установления ассоциативной связи между карточками экспертов и требованиями документации, между пунктами плана тестирования и карточками экспертов. После этого составление итоговых отчетов производится автоматически — они порождаются на основе установленных связей. Основные отчеты: «о покрытии пунктов плана тестирования» (все пункты плана должны быть покрыты), «о реализации заявленной функциональности» (всем требованиям документации должны соответствовать карточки), «о незаявленной функциональности» (карточки без соответствия с требованиями документации) и т. д.

3.4 Понятийно-терминологическая сеть, экспертно-аналитическая система

Ясно, что в сложной системе, когда выделяются сотни (или более) требований документации и создаются сотни сообщений экспертов, процедура установления связи между ними является неочевидной. Здесь могла бы помочь стандартная техника поисковых систем, но в условиях одновременной работы нескольких (многих) экспертов, которые создают краткие сообщения, используя каждый свою лексику, требуется учет синонимов, иерархическое расширение запроса по специальной терминологии. Поэтому в состав комплекса «Тест» включен компонент поддержки понятийно-терминологической сети (информационно-поискового тезауруса), включающий в настоящее время несколько тысяч терминов из предметной области функциональности ПО в средах Unix и Windows. Имеется средство автоматизированного набора терминологии по текстам предметной области (исследуемой специфичной задачи).

Автоматически для всех текстовых фрагментов, с которыми оперирует «Тест» (требования документации, сообщения экспертов) создается морфологический и терминологический поисковый образы. Пункты плана тестирования описываются логической формулой над понятиями понятийной сети.

Средства документирования, использующие информационно-поисковую систему, основанную на понятийно-терминологической сети, образуют «экспертно-аналитическую подсистему» комплекса «Тест». Ее применение дает возможность значительно облегчить работу экспертов по документированию и управляемости процесса тестирования.

Например, достаточно описать пункт плана тестирования как «Ошибки» соответствующей формулой с расширением от понятия «Ошибка» — автоматически будут найдены сообщения экспертов, где обсуждаются любые виды ошибок — «переполнение буфера», «бесконечный цикл» и т. п.

4 Заключение

Мы описали программный комплекс и методологию тестирования функциональности сложного программного обеспечения. Комплекс представляет собой человеко-машинную систему, где базирующееся на знаниях о предметной области программное обеспечение предназначено как для «усиления» интеллекта отдельного эксперта, так и облегчения обмена информацией в группе экспертов-исследователей кода, а также для облегчения управляемости процесса тестирования.

Литература

[1] Канер С., Фолк Дж., Нгуен Е. К. Тестирование программного обеспечения. Киев: Диасофт, 2000.

[2] Рамбо Дж., Якобсон А., Буч Г. UML. Специальный справочник. Спб.: Питер, 2002.

[3] Щербаков А.Ю. Введение в теорию и практику компьютерной безопасности. М.: Молгачева С. В., 2001.

О перспективах решения задачи обфускации компьютерных программ[12]

Н. П. Варновский, В. А. Захаров, Н. Н. Кузюрин, А. В. Шакуров

Развитие современного системного программирования определяется весьма разнообразными тенденциями, предъявляющими противоречивые требования как к самим компьютерным программам, так и к технологии их проектирования. Эти противоречия, будучи осознанными и должным образом формализованными, становятся побудительным мотивом и источником новых математических и инженерных разработок, которые по существу и определяют направления перспективных исследований в теории программирования. Примером тому может служить задача обфускации программ. Эта задача заключается в разработке механизма, позволяющего совместить такие взаимоисключающие качества компьютерных программ как открытость и защищенность. Открытость программы подразумевает свободный и потенциально неограниченный доступ к тексту программы, дающий возможность проводить с программой произвольные эксперименты, анализ и преобразования. В свою очередь, свойство защищенности программы предполагает невозможность или чрезвычайно высокую трудоемкость извлечения из текста программы той ключевой информации, выходящей за рамки прилагающейся к программе спецификации, руководства для пользователя и т. п., которая позволила бы «осознать» программу, понять концепцию ее устройства и затем при необходимости проводить с этой программой сложные преобразования, связанные с целенаправленным изменением ее структурных и функциональных характеристик.

[12]Работа выполнена при поддержке гранта РФФИ 03-01-00880.

В случае построения надежных и стойких обфускаторов программ можно получить решение многих актуальных задач криптографии и компьютерной безопасности. Отметим лишь некоторые прикладные возможности обфускаторов.

1. *Построение криптосистем с открытым ключом.* Предположим, что в нашем распоряжении имеется криптосистема $[E(x,y), D(x,y)]$ с закрытым ключом K, а также стойкий обфускатор $\mathcal{O}$. Тогда, подставив закрытый ключ K в программу шифрования E и применив обфускирующее преобразование $\mathcal{O}$ к специализированному таким образом шифратору, мы получим в результате криптосистему $[\mathcal{O}(E(K,x)), D(x,y)]$ с открытым ключом $\mathcal{O}(E(K,x))$ и секретным ключом K. Стойкость обфускатора $\mathcal{O}$ служит здесь гарантией того, что противник, располагающий открытым ключом, не сможет извлечь из обфускированного шифратора $\mathcal{O}(E(K,x))$ секретный ключ K. Подобная возможность применения обфускаторов для построения стойких криптосистем с открытым ключом отмечалась в основополагающей работе Диффи и Хэлмана [11].

2. *Построение гомоморфных криптосистем.* Располагая стойким обфускатором $\mathcal{O}$, можно секретно выполнять операции над зашифрованными данными. Предположим, что $[E(x,y), D(x,y)]$ — симметричная криптосистема с закрытым ключом K, и c_1 и c_2 — шифры двух битов b_1 и b_2. Тогда для любой бинарной операции $u * v$ справедливо тождество $E(K, b_1 * b_2) = E(K, D(K,c_1) * D(K,c_2))$. Применив обфускатор $\mathcal{O}$ к программе $E(K, D(K,y_1) * D(K,y_2))$, мы получим программу секретного выполнения бинарной операции над зашифрованными данными. Стойкость обфускатора $\mathcal{O}$ гарантирует невозможность извлечения секретной информации (ключа K) из текста обфускированной программы.

3. *Внесение в программу «водяных знаков».* «Водяной знак» — это специальный признак, позволяющий распознавать подлинность или индивидуальность объекта (программы). Этот признак должен обладать свойствами, препятствующими его подделке, удалению и пр. В частности, «водяной знак» должен обладать свойством неотслеживаемости, благодаря которому он становится «невидим» для противника, не располагающего секретными сведениями о помеченной этим знаком программе. Свойство неотслеживаемости «водяных знаков» может быть достигнуто при помощи обфускации — после того, как в программу был внесен «водяной знак», она подвергается обфускации. Стойкость обфускатора гарантирует неотслеживаемость «водяных знаков» в обфускированной программе. Такая возможность применения обфускаторов описана в [7, 9, 10].

Задача обфускации программ была поставлена в [7]. В этой же работе были приведены некоторые простейшие методы преобразования программ, направленные на сокрытие содержащихся в них структур данных и логической структуры алгоритмов. В дальнейшем появилась целая серия работ [4, 6, 8, 5, 15, 17, 20, 21], продолжающая исследования в этом направлении. Используемые в этих работах обфускирующие преобразования программ включают преобразования логической структуры программ, приведение статических данных к динамической форме, введение фиктивных операторов и процедур, манипуляции с переменными-указателями и др. Аргументы в пользу того, что указанные преобразования действительно затрудняют понимание программ основываются на трудности решения задач анализа программ (см. [13, 14, 18]). Так, например, в [6] показано, что для любой задачи H из PSPACE существует эквивалентное преобразование программы π в такую программу π_H, что размер и быстродействие программы π_H ухудшаются незначительно по сравнению с аналогичными характеристиками программы π, но задача выявления фиктивных операторов в программе π_H становится столь же сложной, как и сама задача H.

Основным недостатком всех этих работ с математической точки зрения является отсутствие достаточно строгой формализации самой задачи обфускации. Вследствие этого трудно оценить, в какой мере стойкими являются предложенные обфускирующие преобразования. В данной заметке мы рассмотрим некоторые варианты формального определения понятия обфускации программ и наметим некоторые перспективные направления решения задачи обфускации программ и оценки стойкости обфускирующих преобразований. Задача обфускации программ заключается в разработке такого компилятора программ (обфускатора) $\mathcal{O}$, который удовлетворяет следующим требованиям.

1. Для всякой программы π применение обфускатора $\mathcal{O}$ к программе π дает программу $\mathcal{O}(\pi)$, обладающую теми же функциональными характеристиками, что и программа π;

2. Программа $\mathcal{O}(\pi)$ значительно более сложна для анализа, понимания и извлечения из ее текста полезной информации, нежели исходная программа π.

Одна из возможных формализаций указанных требований, приведенная в [1] (см. также [12]) такова. Обфускатором $\mathcal{O}$ называется всякий компилятор программ, обладающий следующими свойствами.

1. *Функциональность.* Для всякой программы π программы π и $\mathcal{O}(\pi)$ вычисляют одну и ту же функцию;

2. *Полиномиальные издержки.* Для всякой программы π сложность программы $\mathcal{O}(\pi)$ (ее размер, сложность по времени, объему памяти и др.) полиномиально зависит от сложности программы π;

3. *Свойство «виртуального черного ящика».* Для всякого полиномиального вероятностного алгоритма A существует такой полиномиальный вероятностный алгоритм T, что выполняется неравенство
$$|\mathsf{P}\{A(O(\pi)) = 1\} - \mathsf{P}\{T^{\pi}(1^{|\pi|}) = 1\}| \leqslant \mathrm{neg}(|\pi|).$$

Последнее означает, что всякий анализ обфускированной программы на основе ее текста не более плодотворен, чем проведение тестовых испытаний программы без доступа к ее тексту. На основании этого определения в [1] была доказана следующая теорема.

Теорема 1 ([1]). *Не существует универсального обфускатора, обладающего свойством «виртуального черного ящика».*

Как показано в [1], этот результат остается справедливым даже в том случае, когда на универсальный обфускатор не налагается никаких ограничений, связанных с его эффективностью.

Вместе с тем такое решение задачи об обфускации программ нельзя признать окончательным по следующим причинам. Во-первых, полиномиальное увеличение издержек обфускации (увеличение размеров программы, снижение ее быстродействия и пр.) во многих случаях совершенно недопустимо, поскольку при этом стирается грань между принципиально различными алгоритмами решения одной и той же задачи. Так, например, вряд ли возможно признать программу π_1, основанную на стандартном алгоритме умножения целых чисел, результатом обфускации программы π_2, использующей быстрый алгоритм умножения Штрассена, несмотря на то, что программа π_1 вычисляет ту же самую функцию с полиномиальным замедлением по времени и не содержит никакой информации об устройстве программы π_2. Во-вторых, нет никакой необходимости стараться скрыть одновременно все свойства программы от произвольного противника. Как правило, интерес вызывает лишь весьма ограниченный набор свойств, определяющий специфику программы и принцип ее функционирования (например, средства идентификации пользователя, специальные константы, логическая структура программы и т. п.). При этом сама обфускация программы может быть признана успешной, если в результате ее применения значительно понизится эффективность применения существующих в настоящее время средств декомпиляции программ. На основании этого можно предложить следующие альтернативные определения стойкости обфускирующих преобразований.

Свойство нулевой информации о заданном предикате. Рассматривается ансамбль программ $\mathcal{H} = \{(S_n, D_n)\}$, в котором на каждом множестве программ S_n задано распределение вероятностей D_n. Секретное свойство программ задается предикатом P, определенном на множестве всех программ ансамбля $\mathcal{H}$. Тогда на каждой выборке S_n предикат P представляет собой случайную величину $\xi_{P,n}$. Противник $\mathcal{E}$ может быть представлен множеством полиномиальных вероятностных алгоритмов A, принимающих на вход тексты программ ансамбля и вычисляющих один бит. Результат работы алгоритма A на каждой выборке S_n также представляет собой случайную величину $\xi_{A,S,n}$. Тогда будем говорить, что обфускатор $\mathcal{O}$ *выдает нулевую информацию о свойстве* P *на ансамбле программ* $\mathcal{H}$, если выполняется соотношение
$$\mathrm{Inf}(\xi_{P,n}, \xi_{A,\mathcal{O}(S),n}) \leqslant \mathrm{neg}(n),$$
где $\mathrm{Inf}(\alpha, \beta)$ обозначает взаимную информацию случайных величин α и β. Описанный подход к определению стойкости обфускаторов был успешно применен в [19] для оценки стойкости специального

обфускирующего преобразования для одного ансамбля программ $\mathcal{H}_0$. Каждая программа π из семейства S_n имеет размер n и с вероятностью $1/2$ снабжена простой схемой идентификации пользователей. Секретным свойством P_0 программы ансамбля $\mathcal{H}_0$ служит наличие в программе схемы идентификации пользователей. Были установлены следующие теоремы.

Теорема 2 ([19]). *Если существуют односторонние перестановки, то существует обфускатор $\mathcal{O}$, выдающий нулевую информацию о свойстве P_0 на ансамбле программ $\mathcal{H}_0$.*

Теорема 3 ([19]). *Если существует обфускатор $\mathcal{O}$, выдающий нулевую информацию о свойстве P_0 на ансамбле программ $\mathcal{H}_0$, то существуют односторонние функции.*

Предложенный подход может быть использован для теоретической оценки стойкости обфускаторов.

Свойство противодействия алгоритмам статического анализа. Для оценки стойкости обфускаторов можно предполагать также, что возможности противника ограничиваются лишь конечным набором алгоритмов статического анализа программ, позволяющих распознавать свойства вычислений программ без проведения тестовых экспериментов с программами [16]. Большинство алгоритмов статического анализа работают по следующему принципу. Для каждой программы π имеется некоторая априорная совокупность сведений $K_0(\pi)$ о ее свойствах. Работа алгоритма статического анализа заключается в уточнении этих априорных сведений. При этом пространство таких сведений представляет собой полурешетку с максимальным элементом $K_0(\pi)$. Результатом работы алгоритма статического анализа A на программе π является некоторый элемент $A(\pi)$ указанной полурешетки, $A(\pi) \leqslant K_0(\pi)$. В том случае, если $A(p) = K_0(\pi)$, программа π называется непрозрачной для алгоритма A. Будем говорить, что обфускатор $\mathcal{O}$ *противодействует алгоритму статического анализа* A, если каждая программа π преобразуется в эквивалентную непрозрачную для алгоритма A программу $\mathcal{O}(\pi)$. В [22] показано, что для некоторых алгоритмов статического анализа программ можно построить противодействующие обфускаторы.

Свойство противодействия эквивалентным преобразованиями. Предполагается, что возможности противника восстановить (декомпилировать) исходный текст программы π на основе ее образа $\mathcal{O}(\pi)$ ограничиваются применением последовательности преобразований из некоторого фиксированного множества T допустимых эквивалентных преобразований программ. На множестве программ определяется отношение подобия: в множестве T выделяется подмножество T_0 преобразований подобия, и программы π_1 и π_2 считаются T_0-подобными, если одну из них можно получить из другой в результате применения некоторой конечной последовательности преобразований из множества T_0. Будем говорить, что обфускатор $\mathcal{O}$ обладает свойством *противодействия эквивалентным преобразованиям из множества T по отношению T_0-подобия*, если для всякой программы π длина кратчайшей последовательности эквивалентных преобразований, в результате применения которой программа $\mathcal{O}(\pi)$ преобразуется в программу π', T_0-подобную программе π, не может быть ограничена снизу никаким полиномом, зависящим от размера программы π. Подобный подход к анализу стойкости обфускирующих преобразований был исследован в [3].

Свойство увеличения метрики сложности программ. Известно, что для оценки сложности программы можно использовать систему мер, позволяющих ранжировать программы по некоторым критериям качества. К числу таких мер относятся объемные метрики, оценивающие размер программы (число строк, переменных, операторов, функций и пр.), топологические метрики, оценивающие сложность структур управления (цикломатические числа графов управляющих структур программ) и потоков данных (меры Овиеда, Тая, Мак-Клура и др.). В [23] приведен обзор метрической теории программ. На основе таких метрик в [2, 4] была введена композитная мера *цены обфускирующего преобразования*, зависящая от размера программы и размера недостижимых фрагментов программы, а также мера *усложнения программы*, зависящая от сложности графа зависимости по данным. Эти метрики могут быть использованы на практике для приближенной оценки качества обфускации программ.

Литература

[1] Barak B., Goldreich O., Impagliazzo R., Rudich S., Sahai A., Vedhan S., Yang K. On the (Im)possibility of obfuscating programs. CRYPTO'01 - Advances in Cryptology, Lecture Notes in Computer Science, **2139**, 2001, p. 1–18.

[2] Чернов А. В. Анализ запутывающих преобразований программ, В сб.: «Труды Института системного программирования РАН», под ред. В. П. Иванникова. М.: ИСП РАН, 2002, с. 7–37.

[3] Chernov A. A new program obfuscation method, In: Proc. of Int. Workshop on Program Understanding, Novosibirsk, 2003, p. 70–80.

[4] Чернов А. В. Об одном методе маскировки программ, В сб.: «Труды Института системного программирования РАН», под ред. В. П. Иванникова. М.: ИСП РАН, 2003, с. 85–119.

[5] Cho W., Lee I., Park S. Against intelligent tampering: software tamper resistance by extended control flow obfuscation, In: Proc. of the World Multiconference on Systems, Cybernetics and Informatics, 2001.

[6] Chow S., Gu Y., Johnson H., Zakharov V. An approach to the obfuscation of control flow of sequential computer programs. Information Security Conference, Lecture Notes in Computer Science, **2200**, 2001, p. 144–156.

[7] Collberg C., Thomborson C., Low D. A taxonomy of obfuscating transformations, Tech. Report N 148, Dept. of Computer Science, Univ. of Auckland, 1997.

[8] Collberg C., Thomborson C., Low D. Manufacturing cheap, resilient and stealthy opaque constructs. In: Proc. of the Symposium on Principles of Programming Languages, 1998, p. 184–196.

[9] Collberg C., Clark D., Thomborson C. Software Watermarking: Models and Dynamic Embeddings. In: Proc. of the Symposium on Principles of Programming Languages, 1999, p. 311–324.

[10] Collberg C., Thomborson C. Watermarking, Tamper-Proofing, and Obfuscation - Tools for Software Protection. IEEE Transactions on Software Engineering, **28**, No. 6, June 2002.

[11] Diffie W., Hellman M. E. New directions in cryptography. IEEE Transactions in Information Theory, **22**, 1976, p. 644–654.

[12] Hada S. Zero-knowledge and code obfuscation. ASIACRYPT'2000 — Advances in Cryptology, 2000.

[13] Horwitz S. Precise flow-insensitive may-alias analysis is NP-hard. ACM Transactions on Programming Languages and Systems, **19**, N 1, 1997, p. 1–6.

[14] Landi W. Undecidability of static analysis. ACM Letters on Programming Languages and Systems, **1**, N 4, 1992, p. 323–337.

[15] Linn C., Debray S. Obfuscation of executable code to improve resistance to static disassembly. In: Proc. of the 10th. ACM Conference on Computer and Communications Security (CCS 2003), Oct. 2003.

[16] Nielson F., Nielson H. R., Hankin C. Principles of program analysis. Springer-Verlag, 1999, 450 p.

[17] Ogisso T., Sakabe Y., Sochi M., Miyaji A. Software obfuscation on a theoretical basis and its implementation. IEEE Transactions on Fundamentals, E86-A(1), 2003.

[18] Ramalingam G. The undecidability of aliasing. ACM Transactions on Programming Languages and Systems, **16**, N 5, 1994, p. 1467–1471.

[19] Varnovsky N. P., Zakharov V. A. On the possibility of provably secure obfuscating programs, In: Proc. of the 5th Int. Conference Perspectives of System Informatics (PSI'03), 2003, p. 71–78.

[20] Wang C., Hill J., Knight J. Davidson J. Software tamper resistance: obstructing static analysis of programs. Tech. Report N 12, Dep. Of Comp. Sci., Univ. of Virginia, 2000.

[21] Wroblewski G. General method of program code obfuscation, In: Proc of the Int. Conference on Software Engineering Research and Practice (SERP), 2002, p. 153–159.

[22] Иванов К. С., Захаров В. А. О противодействии некоторым алгоритмам статического анализа программ. Наст. сборник.

[23] Черноножкин С. К. Меры сложности программ (обзор). В сб.: Системная информатика, вып. 5, 1997, с. 188–228.

Обеспечение информационной безопасности систем на программной платформе ос2000

В. Б. Бетелин, В. А. Галатенко, А. Н. Годунов, А. И. Грюнталь

Аннотация

Обеспечение информационной безопасности (ИБ) аппаратно-программной платформы – необходимое условие построения защищенных систем. В докладе рассматривается понятие аппаратно-программной платформы, дается краткая характеристика операционной системы реального времени ос2000, анализируется ее соответствие требованиям «Общих критериев». Описываются особенности современных информационных систем с точки зрения ИБ, рассматриваются теоретические и практические аспекты обеспечения их информационной безопасности с применением решений на отечественной аппаратно-программной платформе.

1 Понятие аппаратно-программной платформы и ее роль в обеспечении информационной безопасности

Любую информационную систему (ИС) можно подразделить на две составляющие:

- прикладную;
- инфраструктурную.

Первая реализует функциональность, присущую только данной системе или узкому классу систем, предоставляя прикладные сервисы. Вторая является фундаментом первой, предоставляя для нее системные сервисы и обеспечивая наличие определенных свойств, необходимых для успешной разработки, эксплуатации и модернизации. Важнейшим из таких свойств является информационная безопасность (ИБ), которая даже для больших, сложных систем зависит не только и не столько от применения каких-то специфических средств, сколько от инфраструктуры, основу которой составляют аппаратно-программные платформы – совокупность аппаратного обеспечения и операционных систем (ОС).

К сожалению, в плане информационной безопасности аппаратно-программные платформы оказываются не только самым важным, но и слабым звеном, как с формальной, так и с содержательной точек зрения. Формальное условие сертификации аппаратно-программного обеспечения по требованиям безопасности практически невыполнимо в силу доминирования зарубежных аппаратных и программных продуктов, а также в силу их высокой сложности. Та же сложность является основным препятствием и на пути обеспечения реальной безопасности. Известны ошибки в микропроцессорах компании Intel. Постоянно обнаруживаются новые серьезные уязвимости в универсальных операционных системах. Все это заставляет еще раз проанализировать существующие подходы к обеспечению информационной безопасности аппаратно-программных платформ, сопоставить их с современным уровнем информационных технологий, попытаться предложить новые решения.

В Научно-исследовательском институте системных исследований Российской Академии наук (НИИСИ РАН) ведется разработка базового аппаратного и программного обеспечения, создана отечественная аппаратно-программная платформа. Тема настоящего доклада – программная составляющая этой платформы: операционная система реального времени ос2000 и связанные с ней формальные и содержательные проблемы обеспечения информационной безопасности.

2 Краткая характеристика ос2000

С возможностями ос2000 можно ознакомиться по статье [1]. Для нас существенны следующие позитивные (описывающие, то, что присутствует в ОС) свойства:

- следование международному стандарту POSIX 1003.1 [2] в части, касающейся программного интерфейса к средствам ввода/вывода, механизмам реального времени, средствам протоколирования;
- следование международному стандарту языка Си [3] в плане поддержки среды времени выполнения;
- следование спецификациям семейства протоколов TCP/IP, наличие реализации протоколов прикладного уровня: FTP (клиент), TELNET (сервер и клиент), NFS (сервер и клиент).
- развитые средства конфигурирования, возможность отбора только тех элементов ОС, которые необходимы для работы приложений.

Не менее существенны и негативные свойства (описывающие то, что в ос2000 отсутствует):

- отсутствие поддержки двух режимов работы микропроцессоров, использование только привилегированного режима;
- отсутствие поддержки механизма виртуальной памяти и деления адресного пространства на системное и пользовательское;
- отсутствие понятия процесса, функционирование всех потоков в одном адресном пространстве;
- отсутствие понятия пользователя и, как следствие, отсутствие средств разграничения доступа.

Основное следствие позитивных свойств – мобильность программного обеспечения (ПО), следующего стандартам, в сочетании с достаточно широким спектром функциональных возможностей.

Следствие негативных свойств – простота и компактность ОС, минимизация числа возможных ошибок и уязвимостей. Отметим в этой связи аккуратность реализации стека протоколов TCP/IP. Система анализа защищенности Nessus [4] обнаружила лишь три потенциальные уязвимости с низким уровнем риска: поддержка небезопасного протокола TELNET, предсказуемость идентификаторов IP-пакетов, обслуживание запросов временных штампов в протоколе ICMP. Для сравнения: анализ конфигурации с ОС Solaris 2.5 выявил 18 уязвимостей с высоким уровнем риска (и несколько десятков других со средним у низким уровнями), в том числе серьезные проблемы в реализации протоколов FTP и TELNET. Конечно, версия Solaris 2.5 довольно старая, но, с другой стороны, предыстория у нее значительно богаче, чем у ос2000.

В то же время, отсутствие в ос2000 поддержки двух базовых защитных механизмов – двух режимов работы микропроцессоров и механизма виртуальной памяти – заставляет искать новые решения в области архитектурной безопасности, а отсутствие средств идентификации/аутентификации и разграничения доступа создает по крайней мере формальные проблемы при попытке применить существующие Руководящие документы Гостехкомиссии России или аналогичные ведомственные нормативные документы.

В последующих разделах мы детально рассмотрим отмеченные выше и некоторые другие проблемы, предложим подходы к их решению.

3 Особенности современных информационных систем с точки зрения информационной безопасности

За понятием «современная информационная система» скрывается целый спектр конфигураций – от глубоко встроенных до корпоративных с территориально распределенными компонентами. Тем не менее, можно выделить некоторые общие свойства и тенденции, присущие конфигурациям всех видов.

Среди основных аспектов информационной безопасности – доступности, целостности и конфиденциальности – все более высокий приоритет получает доступность. Это верно не только для встроенных систем, систем управления и т.п., для которых в первую очередь создавалась ос2000, но и для Интернет-порталов, систем электронной коммерции и многих других. Не случайно наибольшее распространение в последнее время получили именно атаки на доступность, а наибольшее внимание общественности привлекли вызванные ими перерывы в работе крупнейших Интернет-порталов.

В состав современных ИС входят компоненты на специализированных аппаратно-про-граммных платформах, с ограниченными ресурсами. Это могут быть как встроенные (под)системы, так и потребительские устройства с поддержкой сетевого доступа. В качестве специализированной программной платформы для подобных компонентов необходимо использовать операционную систему класса ос2000. Универсальные ОС не годятся по двум причинам:

- им требуется слишком много ресурсов;
- им требуется квалифицированное администрирование, которое для потребительских устройств не может быть обеспечено.

Для подобных компонентов важна не только доступность, но также целостность программ и данных, поскольку возврат к корректной конфигурации или невозможен (отсутствуют резервные копии), или очень сложен (требуется вмешательство специалистов).

Современные ИС включают разнородные серверные компоненты и строятся в многоуровневой архитектуре. Для того, чтобы скрыть разнородность и архитектурную сложность от пользователей, в качестве информационного концентратора используют Web-серверы, скрывающие за универсальными локаторами ресурсов (URL) вызовы информационных сервисов с определенными параметрами. Пользователи применяют единообразный навигационный интерфейс, получая результаты запросов в виде HTML- (или, в более общем случае, XML-) страниц.

В многоуровневой архитектуре ни одна серверная операционная система не осуществляет разграничение доступа в традиционном понимании. Разграничением (и администрированием) доступа пользователей занимается Web-сервер, но, во-первых, он делает это на прикладном уровне (удаленные пользователи, разумеется, в серверной ОС не регистрируются), а, во-вторых, он управляет доступом к ресурсам, ему (и его ОС) не принадлежащим, физически располагающимся на других серверах; последние, в свою очередь, ничего не знают об удаленных пользователях. Таким образом, главными задачами серверной ОС являются поддержка сетевого взаимодействия, сохранение целостности локальной конфигурации, обеспечение безопасного администрирования и невозможности локального обхода защитных средств, а также протоколирование.

В многоуровневой архитектуре строятся не только сами ИС, но и их защитные средства. Для некоторых из них реализация в рамках отдельных узлов сети уже стала привычной (достаточно упомянуть межсетевые экраны, опорные узлы виртуальных частных сетей и виртуальных локальных сетей); для других (таких, как сервер аутентификации, сервер хранения и анализа регистрационной информации) подобная реализация – вероятно, дело ближайшего будущего. В соответствующих профилях защиты неизменно присутствует предположение безопасности следующего вида: «В программной платформе присутствуют только те механизмы и функции, которые необходимы защитному средству». Удалить все лишнее из универсальной ОС – задача едва ли выполнимая; практичнее и безопаснее использовать компактную, конфигурируемую платформу класса ос2000. Налицо ситуация, которая кажется парадоксальной, но является таковой лишь на первый взгляд: универсальная, защищенная ОС – не лучшая программная платформа для защитных средств.

Несомненной тенденцией развития ИС является усиление сетевой связности, как внутренней, так и внешней. К открытым сетям подключают все большее число систем, в том числе встроенных систем и систем управления. Можно назвать две основные причины усиления внешней связности:

- внешние сети могут предоставить информационные сервисы, придающие встроенным системам и системам управления новое качество (глобальное время, глобальное позиционирование и т.п.);
- внешняя связность позволяет контролировать и администрировать системы практически из любой точки, что важно для территориально распределенных конфигураций.

Очевидно, внешняя связность создает дополнительные угрозы безопасности. Известный специалист Б. Шнейер в своей статье [5], посвященной безопасности систем управления, в качестве основной рекомендации предлагает держать критически важные комплексы подальше от Интернет (две другие рекомендации бесспорны: совершенствовать протоколы для усиления безопасности и не паниковать по поводу угроз, поскольку риск не так уж велик). Теоретически это правильно, но, к сожалению, на практике доминирует противоположная тенденция. Это значит, что даже для специализированных систем с ограниченными ресурсами необходимо решать универсальные проблемы сетевой безопасности.

4 Теоретические основы обеспечения информационной безопасности современных ИС

4.1 Интерпретация «Оранжевой книги» для сетевых конфигураций

Теоретические основы оценки безопасности сетевых систем были заложены в «Интерпретации «Оранжевой книги» для сетевых конфигураций» [6]. По определению, доверенная (защищенная) сетевая система должна обеспечивать такое распределение защитных механизмов, чтобы общая политика безопасности проводилась в жизнь несмотря на уязвимость коммуникационных путей и на параллельную, асинхронную работу компонентов. Не существует прямой зависимости между уровнями безопасности отдельных компонентов и уровнем безопасности всей сетевой конфигурации. Например, в результате объединения двух компонентов, один из которых сам не обеспечивает протоколирование действий пользователя, но передает необходимую информацию другому компоненту, который и ведет протокол, образуется сеть, удовлетворяющая требованию подотчетности, несмотря на слабость компонента.

«Интерпретация...» предусматривает различные варианты распределения механизмов управления доступом. В частности, компоненты, закрытые от прямого доступа пользователей, могут вообще не содержать подобных механизмов. В принципе возможен централизованный контроль доступа, когда решения принимает специальный сервер авторизации. Возможен и смешанный вариант, когда сервер авторизации разрешает соединение двух хостов, а дальше используются локальные механизмы хоста, содержащего объект доступа. Аналогично, идентификация и аутентификация может производиться как централизованно (соответствующим сервером), так и локально – той системой, с которой пользователь непосредственно взаимодействует. Возможна передача идентификационной и аутентификационной информации между хостами, чтобы избавить пользователя от многократной аутентификации. В общем случае компоненты, поддерживающие лишь часть необходимых сервисов безопасности, должны обладать программными и/или протокольными интерфейсами, чтобы получить недостающие им сервисы от других компонентов, выполняющих экранирующие функции. Один компонент может экранировать сколь угодно большое число других; прямой доступ к незащищенным компонентам (минуя экран) должен быть исключен.

В «Интерпретации...» учтена динамичность сетевых конфигураций. Предусматривается наличие средств проверки подлинности и корректности функционирования компонентов перед их включением в сеть, наличие протокола взаимной проверки компонентами доступности и корректности функционирования друг друга, реализация средств оповещения администратора о неполадках в сети. Сетевая конфигурация должна быть устойчива к отказам отдельных компонентов или коммуникационных путей.

Сетевой сервис перестает быть доступным, когда пропускная способность коммуникационных каналов падает ниже минимально допустимого уровня или сервис не в состоянии обслуживать запросы. Удаленный ресурс может стать недоступным и вследствие нарушения равноправия в обслуживании пользователей. Доверенная (защищенная) система должна быть в состоянии обнаруживать ситуации недоступности, уметь возвращаться к нормальной работе и противостоять атакам на доступность. Для обеспечения высокой доступности могут применяться следующие защитные меры:

- внесение в конфигурацию той или иной формы избыточности (резервное оборудование, запасные каналы связи и т.п.);
- наличие средств реконфигурирования для изоляции и/или замены узлов или коммуникационных каналов, отказавших или подвергшихся атаке на доступность;
- рассредоточенность сетевого управления, отсутствие единой точки отказа;
- наличие средств нейтрализации отказов (обнаружение отказавших компонентов, оценка последствий, восстановление после отказов);
- выделение подсетей и изоляция групп компонентов друг от друга.

Продолжая тему доступности, упомянем класс функциональности FAV из «Гармонизированных критериев Европейских стран» [7]. В разделе «Надежность обслуживания» описания этого класса специфицируется, что система должна восстанавливаться после отказа отдельного компонента таким

образом, чтобы все критически важные функции оставались постоянно доступными. То же должно быть верно для активизации восстановленного компонента, причем после этого система возвращается в состояние, устойчивое к одиночным отказам. Независимо от уровня загрузки должно гарантироваться время реакции на определенные события и отсутствие тупиков.

4.2 Критерии оценки безопасности информационных технологий (международный стандарт ISO/IEC 15408)

То, что к безопасности информационных систем нужно предъявлять не априорные требования, а требования, вытекающие из специфики ИС, было показано еще в 1991 году, в упоминавшихся выше «Гармонизированных критериях Европейских стран». Это фундаментальное положение получило дальнейшее развитие в международном стандарте ISO/IEC 15408-1999, более известном как «Общие критерии» [8, 9, 10, 11]. Набор функциональных требований «Общих критериев» не претендует на полноту – при необходимости могут добавляться новые. Однако все требования, вошедшие в стандарт (кроме, быть может, класса «приватность»), несомненно, должны учитываться при проектировании, реализации и оценке ИС. Подчеркнем, что в качестве объектов оценки мы рассматриваем законченные информационные системы, а не отдельные компоненты.

Рассмотрим подробнее два класса требований, наиболее важных в контексте данной статьи – защиту функций безопасности и использование ресурсов. Аппаратно-программная платформа должна обеспечивать следующие виды защиты функций безопасности:

- Физическая защита (пассивная и активная). Пассивная защита должна обнаруживать физические вторжения и информировать о них администратора безопасности. Активная защита должна противодействовать действиям нарушителя, например, выключать устройства, чтобы сохранить конфиденциальность данных.
- Тестирование базовой абстрактной машины, самотестирование функций безопасности. Платформа должна комплектоваться тестами, выполняющимися при старте, с определенной периодичностью или в соответствии с иной дисциплиной.
- Разделение доменов. Должен поддерживаться отдельный домен для выполнения функций безопасности, который защищает их от вмешательства и искажения недоверенными субъектами.
- Невозможность обхода. Политика безопасности должна применяться прежде, чем разрешается выполнение каких-либо иных операций.
- Безопасность при сбоях, безопасное восстановление (ручное или автоматическое), согласованность данных, используемых разными функциями безопасности, при сбоях и после восстановления, синхронизация состояний функций безопасности.
- Конфиденциальность, целостность и доступность данных, экспортируемых функциями безопасности.

Класс «Использование ресурсов» содержит требования, направленные на поддержание доступности ресурсов, таких как такты процессора, оперативная память или сетевая связность. Эти требования подразделяются на три семейства:

- Отказоустойчивость. При сбоях должна сохраняться работоспособность избранных или всех функций объекта оценки.
- Приоритетность обслуживания. Субъектам должны назначаться приоритеты, доступ к ресурсам должен обеспечиваться на основе назначенных приоритетов.
- Распределение ресурсов. Должно обеспечиваться выделение некоторого минимального количества ресурсов, должны отслеживаться лимиты на использование ресурсов.

5 Возможная архитектура защищенных систем с компонентами на платформе ос2000

5.1 Анализ ос2000 на соответствие требованиям безопасности «Общих критериев»

5.1.1 Соответствие требованиям доверия безопасности

Коллектив разработчиков ос2000 ориентировался и ориентируется на комплекс мер доверия безопасности, описанных в «Общих критериях», причем имеются в виду не только элементы действий разработчика, но и элементы представления и содержания свидетельств. В соответствии с требованиями, сформулированными в классе ADV «Разработка», поддерживаются несколько уровней представления – от функциональной спецификации до представления реализации. Неформальными методами демонстрируется соответствие между этими уровнями. Выполняются также требования к внутренней структуре (модульность, разбиение на уровни, минимизация сложности).

Представляется крайне важным охват всего жизненного цикла, который достигается выполнением требований класса ALC «Поддержка жизненного цикла». Прежде всего, это обеспечение безопасности разработки, которая достигается комплексом программных и организационных мер. Далее, для разработки используются только стандартизованные, согласованные между собой инструментальные средства в сочетании со стандартизованной моделью жизненного цикла. Наконец, отработана процедура устранения недостатков, выявляемых в процессе эксплуатации.

Тестирование выполняется в соответствии с требованиями класса ATE. Проводится функциональное тестирование, обосновывается достаточные глубина и покрытие тестового набора.

Выполнены и важные с практической точки зрения требования класса ADO «Поставка и эксплуатация». Потребитель получает именно то, что заказал, без каких-либо несанкционированных модификаций, а установка, генерация и запуск подробно описаны в соответствующих Руководствах.

Важным техническим элементом является управление конфигурацией, выполняемое в соответствии с требованиями класса ACM. Частичная автоматизация управления конфигурацией и полное покрытие подтверждают соответствие между функциональными спецификациями и реализацией, обеспечивают отслеживание изменений, предохраняют от несанкционированных (в том числе случайных) модификаций.

Документация выполнена в соответствии с национальными стандартами, а также требованиями класса AGD «Руководства». Наличие Руководств программиста и системного программиста позволяет отделить пользовательские аспекты от административных, упростить понимание документации и, соответственно, облегчить применение, сделав тем самым ос2000 более безопасной.

В целом группа мер, реализуемых в рамках разработки ос2000, соответствует четвертому оценочному уровню доверия «Общих критериев». Тем самым достигается достаточная для предполагаемых областей применения степень доверия информационной безопасности платформы.

5.1.2 Соответствие функциональным требованиям

Система на платформе ос2000, рассматриваемая изолированно, удовлетворяет следующим функциональным требованиям:

- требования к генерации регистрационной информации (семейство FAU_GEN) и выбору регистрируемых событий (семейство FAU_SEL) класса FAU «Аудит безопасности»);
- требования тестирования и самотестирования (класс FPT «Защита функций безопасности»);
- требования приоритетности обслуживания (класс FRU «Использование ресурсов») применительно к доступу к процессору.

Таким образом, ос2000 как изолированная система практически лишена защитных средств. Важнейшие требования разделения доменов и невозможности обхода могут быть выполнены в рамках сетевых конфигураций, когда на одном узле (с ос2000 в качестве платформы) функционирует один информационный или защитный сервис, локальный доступ к узлу невозможен, а сетевой производится

по определенному, контролируемому протоколу. Пример подобного выделенного домена – подключенный к сети компьютер с графическим сервером X Window на платформе ос2000 (и с отключенной поддержкой других сервисов прикладного уровня).

Коротко правило разделения доменов для ос2000 можно сформулировать как «один домен на узел сети с одним сервисом» или как «разнесение доменов по узлам сети». При современной дешевизне аппаратуры подобный подход не является расточительным, однако проблемой может стать нехватка полосы пропускания сети, если разные сервисы, обычно реализуемые в рамках одной многопроцессной системы, интенсивно обмениваются данными. Здесь решением может стать объединение одноплатных компьютеров с общей шиной, обладающей высокой пропускной способностью.

Для выполнения других требований безопасности необходимо применение сетевых конфигураций с достаточным набором экранирующих сервисов. К рассмотрению соответствующих архитектур мы и переходим.

5.2 Архитектура с экранирующими сервисами безопасности

Сетевые конфигурации с узлами на платформе ос2000, предназначенные для обслуживания пользователей, могут быть защищены с помощью межсетевых экранов, средств поддержки виртуальных частных сетей и Web-серверов и серверов аудита. При этом архитектура сети должна быть такой, чтобы выполнялось требование невозможности обхода защитных средств.

Межсетевой экран берет на себя поддержку понятия пользователя, осуществляя идентификацию/аутентификацию и разграничение доступа к сетевым сервисам. Кроме того, на нем могут быть реализованы функции виртуальной частной сети, что обеспечит конфиденциальность и целостность потоков данных. Дополнительным защитным рубежом может служить Web-сервер, осуществляющий более тонкое, чем межсетевой экран, разграничение доступа и унифицирующий интерфейс к сервисам внутренней сети. Сервер аудита берет на себя хранение регистрационной информации, выявление подозрительной активности и реагирование на нее.

В рамках описанной конфигурации могут быть выполнены все классы функциональных требований «Общих критериев». Внутренняя сеть может быть реализована как набор аппаратных модулей, заключенных в одном конструктиве; там же могут располагаться экранирующие защитные сервисы. В результате получится законченное защищенное аппаратно-программное решение.

6 Заключение

POSIX-совместимая операционная система реального времени ос2000 сочетает достоинства компактности и простоты с богатством предоставляемых функциональных возможностей. Это делает ее пригодной для использования в качестве программной платформы не только в системах жесткого реального времени, но и в произвольных системах с ограниченными ресурсами (например, в потребительских устройствах).

Формальная оценка безопасности отдельных систем на платформе ос2000 не имеет смысла. Необходимо рассматривать законченные сетевые конфигурации, обеспечивающие разделение доменов за счет разнесения по узлам сети и использующие экранирующие сервисы, такие как межсетевое экранирование. Реализация сервисов безопасности на платформе ос2000 позволяет получить решения, допускающие сертификацию на всех уровнях – от аппаратного до прикладного.

Литература

[1] Безруков В. Л., Годунов А. Н., Назаров П. Е., Солдатов В. А., Хоменков И. И. Введение в ос2000. – В сб. «Вопросы кибернетики. Информационная безопасность. Операционные системы реального времени. Базы данных» под ред. чл.-корр. РАН В. Б. Бетелина. – М.: НСК РАН, 1999, с. 76–106.

[2] POSIX: Information Technology – Portable Operating System Interface (POSIX) – Part 1: System Application Program Interface (API) [C language]. ISO/IEC 9945-1.

[3] International Standard ISO/IEC 9899:1990. Programming languages – C.

[4] Система анализа защищенности Nessus. – http://www.nessus.org.

[5] Schneier B. Embedded Control Systems and Security. – CryptoGram, July 15, 2002. http://www.counterpane.com/crypto-gram-0207.html#1

[6] National Computer Security Center. Trusted Network Interpretation. – NCSC-TG-005, 1987.

[7] Information Technology Security Evaluation Criteria (ITSEC). Harmonised Criteria of France – Germany – the Netherlands – the United Kingdom. – Department of Trade and Industry, London, 1991.

[8] Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model. – ISO/IEC 15408-1.1999.

[9] Information technology – Security techniques – Evaluation criteria for IT security – Part 2: Security functional requirements. – ISO/IEC 15408-2.1999.

[10] Information technology – Security techniques – Evaluation criteria for IT security – Part 3: Security assurance requirements. – ISO/IEC 15408-3.1999.

[11] Трубачев А. П., Долинин М. Ю., Кобзарь М. Т., Сидак А. А., Сороковиков В. И. Оценка безопасности информационных технологий. Под общ. ред. В. А. Галатенко. – М.: СИП РИА, 2001. – 356 с.

Опыт разработки и перспективы применения безопасных систем на базе защищенной ОС

П. Д. Зегжда

1 Задача защищенной ОС «Феникс»

Основная задача ЗОС «Феникс» — защита систем обработки информации путем обеспечения безусловного выполнения правил системной политики безопасности для всех информационных взаимодействий с помощью средств контроля и управления доступом. Главной архитектурной особенностью ЗОС «Феникс» является оригинальный подход к реализации контроля и управления доступом, основанный на концепциях «информационного ресурса» и «универсального интерфейса доступа», позволяющий распространить общесистемную политику безопасности на все операции доступа независимо от способа его осуществления и природы защищаемых информационных ресурсов.

Информационный ресурс — это любая способная участвовать в отношениях доступа в качестве источника либо приемника информации абстрактная сущность, обладающая уникальным идентификатором, доступ к которой осуществляется с помощью фиксированного набора операций, работающих с содержанием только одного ресурса и осуществляющих передачу информации только в одном направлении – либо от потребителя к ресурсу, либо обратно.

В качестве защищаемых информационных ресурсов могут выступать файлы, html-страницы, документы, учетные записи пользователей, сетевые соединения и т. п. Защищаемые ресурсы могут быть как локальными, так и сетевыми. Локальные ресурсы принадлежат непосредственно ЗОС «Феникс», тогда так защищаемые сетевые информационные ресурсы — это разделяемые ресурсы других систем, для работы с которыми ЗОС «Феникс» используется в качестве шлюза.

Универсальный интерфейс доступа определяет унифицированный для всех типов информационных ресурсов набор операций, включающий операции доступа к информационному содержанию ресурса, его создания, уничтожения и управления свойствами, в том числе атрибутами безопасности.

Все функции защиты действуют по отношению к контролю взаимодействия субъектов и информационных ресурсов, где субъекты представляют пользователей (локальных или удаленных), а ресурсы

находятся под контролем ЗОС «Феникс». Представление защищаемой информации в виде информационных ресурсов позволяет контролировать все виды информационных взаимодействий, а благодаря использованию универсального интерфейса доступа механизмы защиты ЗОС «Феникс» инвариантны как по отношению к способам осуществления доступа, так и к различным типам информационных ресурсов. Построенные на базе предложенных подходов механизмы защиты позволяют ЗОС «Феникс» решать задачи защиты информации от несанкционированного доступа и соблюдения правил политики безопасности для авторизованных пользователей.

2 Принципы построения ЗОС «Феникс»

В основе концепции построения ЗОС «Феникс» лежат следующие принципы:

1. *Принцип абсолютности* — средства защиты должны быть встроены в систему обработки информации таким образом, чтобы абсолютно все, без исключения, механизмы взаимодействия находились под их контролем;

2. *Принцип инвариантности* — средства защиты должны быть независимы от приложений и логики их функционирования. Все информационные взаимодействия в системе должны иметь форму операций доступа субъектов к объектам, что позволит контролировать их с помощью универсальных механизмов защиты, инвариантных к типу взаимодействий;

3. *Принцип унификации* — множество операций субъектов над объектами, контролируемых средствами безопасности, должно однозначно отображаться на множество отношений доступа, описываемых моделями безопасности. Это обеспечивает универсальность средств защиты и позволяет использовать их без изменения как для реализации различных моделей безопасности, так и для контроля доступа к объектам различной природы;

4. *Принцип рефлексивности* — для того чтобы успешно противостоять деструктивным воздействиям (атакам), необходимо устранить недостатки (уязвимости), присущие современным системам, благодаря которым они подвергаются успешным нападениям;

5. *Принцип генезиса* — для создания системы, лишенной уязвимостей, следует устранить источники их появления, главными из которых являются:

   - отсутствие последовательного подхода при реализации контроля доступа;
   - наличие привилегированных средств, передающих пользователям часть своих полномочий в обход средств защиты;
   - наличие ошибок в реализации средств защиты;

6. *Принцип адекватности* — защищенная система должна обеспечивать эквивалентную реализацию информационных потоков обслуживаемого ею информационного процесса, т. е. допускать существование только тех потоков информации, которые являются частью этого процесса;

7. *Принцип концептуальности* — средства защиты должны осуществлять управление доступом в соответствии с правилами непротиворечивой, формально доказанной модели безопасности.

3 Системная архитектура ЗОС «Феникс»

Поскольку основной задачей ЗОС «Феникс» является обеспечение безопасности, все ее компоненты в той или иной степени участвуют в реализации функций защиты, а архитектура является воплощением предложенных принципов оригинальной технологии создания защищенных систем применительно к операционной системе. Наличие обоснованной архитектуры составляет принципиальное отличие ЗОС «Феникс» от традиционных защищенных систем, представляющих собой доработку систем общего назначения, но сохранивших при этом все недостатки, присущие их архитектуре. Архитектура ЗОС «Феникс» является главным механизмом обеспечения безопасности, поскольку гарантирует всеобъемлющий и непрерывный контроль всех информационных взаимодействий, составляющий основу функционирования всех средств защиты.

Основу ЗОС «Феникс» составляет отвечающая принципу *абсолютности* микроядерная архитектура, предусматривающая единственный способ взаимодействия между компонентами системы с помощью механизма обмена сообщениями, реализованного в микроядре. Встроенные в этот механизм средств защиты пропускают через себя все потоки сообщений, что гарантирует тотальный контроль всех взаимодействий в системе.

В соответствии с принципом *инвариантности* все взаимодействия между компонентами ЗОС «Феникс» осуществляются на основе технологии клиент-сервер, четко регламентирующей роли взаимодействующих сторон не зависимо от его природы, типа и способа осуществления доступа.

В качестве инициатора взаимодействия выступает процесс-клиент, являющийся субъектом, а исполнителем операции является процесс-сервер. Для обозначения предмета осуществления операции используется понятие ресурса как абстрактного представления объекта доступа.

В зависимости от источника происхождения и способа осуществления доступа ресурсы разделяются на аппаратные и информационные.

Аппаратные ресурсы – это ресурсы оборудования вычислительной системы, непосредственно потребляемые процессами (оперативная память, процессорное время, порты ввода-вывода и т. п.) и предоставляемые в их распоряжение микроядром.

Под информационным ресурсом понимается реализованная программными средствами абстракция любого уровня (файл, каталог, сокет, пространство жесткого диска, учетная запись пользователя и т. д.), обладающая уникальным идентификатором, инкапсулированная в некоторый программный объект, предоставляющий фиксированный набор методов для доступа к ней. В ЗОС «Феникс» этот объект реализуется в виде процесса-сервера, который представляет совокупность однотипных ресурсов. Информационные ресурсы, в отношении которых действует политика безопасности, называются объектами.

Процессы-серверы обслуживают запросы процессов-клиентов на использование ресурсов, представляющих средства и возможности системы, путем обслуживания исходящих от них запросов к этим ресурсам. В этой схеме все взаимодействия принимают форму обращения процессов-клиентов к ресурсам, которые обслуживаются процессами-серверами. Поскольку все информационные ресурсы ЗОС «Феникс» находятся под контролем процессов-серверов, обрабатывающих запросы на использование этих ресурсов, поступающие от процессов-клиентов, субъектом доступа всегда является процесс-клиент, представляющий пользователя системы, а объектом — ресурс, поддерживаемый процессом-сервером. Существует единственный механизм взаимодействия — посылка сообщения, содержащего идентификаторы субъекта, объекта и запрашиваемой операции.

В соответствии с принципом *унификации* доступ к информационным ресурсам ЗОС «Феникс» осуществляется с помощью универсальных операций «Унифицированного интерфейса доступа к информационным ресурсам» (УНИДИР). УНИДИР определяет множество универсальных для всех типов информационных ресурсов операций доступа к содержащейся в них информации, создания, уничтожения ресурсов и управления их свойствами. Все операции УНИДИР однозначно отображаются на отношения доступа, регламентируемые моделями безопасности, используемыми в ЗОС «Феникс». Использование УНИДИР является единственным способом осуществления операций над объектами.

4 Архитектура безопасности ЗОС «Феникс»

Микроядерная архитектура, взаимодействие по технологии клиент-сервер, концепция ресурсов и УНИДИР гарантируют всеобъемлющее и непрерывное функционирование средств защиты ЗОС «Феникс», схема функционирования которых приведена на рисунке 1. Средства аутентификации обеспечивают доступ к системе только авторизованных пользователей.

Процесс, выполняющийся от имени пользователя и обладающий его полномочиями, осуществляет доступ к ресурсам ЗОС «Феникс» под обязательным контролем средств идентификации, которые определяют соответствующий ему субъект доступа. Все обращения к информационным ресурсам осуществляется под контролем средств управления доступом, которые обеспечивают выполнение правил политики безопасности при доступе к тем информационным ресурсам, которые являются объектами. Средства контроля за потреблением аппаратных ресурсов ограничивают их потребление в соответствии с полномочиями субъектов и обеспечивают работу остальных средств защиты.

Все операции доступа к ресурсам (как успешные, так и неуспешные) заносятся в протокол аудита.

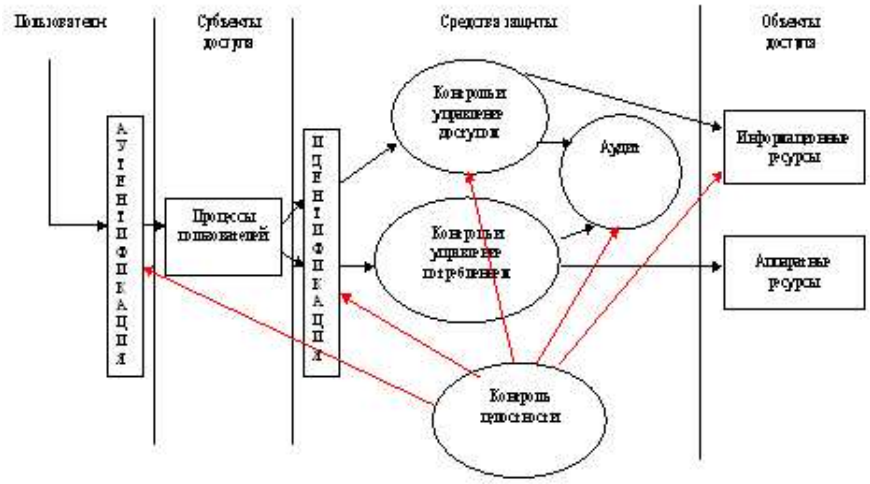


Рис. 1: Схема функционирования средств защиты ЗОС «Феникс».

Средства контроля целостности обеспечивают целостность средств защиты и восстановление целостности информационных ресурсов.

5 Контроль и управление потреблением аппаратных ресурсов

В основе всех информационных ресурсов лежат те или иные ресурсы аппаратной платформы (процессор, оперативная или дисковая память, порты ввода/вывода). Поэтому контроль доступа и управление потреблением на уровне взаимодействия процессов с аппаратурой являются базовыми функциями, обеспечивающими работу всех остальных средств защиты.

Контроль и управление потреблением аппаратных ресурсов ЗОС «Феникс» обеспечивает:

- одновременное функционирование множества процессов и потоков, совместно использующих аппаратные ресурсы;
- невозможность осуществления вмешательства в работу процесса со стороны других процессов или какого-либо воздействия на него;
- невозможность действовать в обход средств контроля и управления доступом;
- невозможность вмешательства в работу средств защиты или нарушения их целостности;
- возможность управления распределением аппаратных ресурсов между пользователями с помощью квот и полномочий.

Средства контроля и управления потреблением аппаратных ресурсов ЗОС «Феникс» включают:

- механизм изоляции адресных пространств процессов, обеспечивающий полную изоляцию адресных пространств всех процессов, основанный на аппаратных возможностях процессора Intel;
- механизм контроля доступа к портам ввода-вывода, запрещающий доступ к ним для процессов пользователя;
- механизм уничтожения остаточной информации, предотвращающий утечки информации при повторном использовании памяти путем ее обнуления при выделении;
- средства учета объемов потребления оперативной памяти и дискового пространства для каждого субъекта;

- средства ограничения объемов аппаратных ресурсов (процессорного времени, оперативной памяти и дискового пространства), потребляемых совокупностью процессов, представляющих одного субъекта, с помощью гибкого механизма квот, устанавливаемых персонально для каждого субъекта;
- механизм планирования распределения процессорного времени, устанавливающий приоритеты процессов в зависимости от атрибутов безопасности субъектов, которых представляют эти процессы.

6 Контроль и управление доступом к информационным ресурсам

Вследствие соблюдения принципа *концептуальности* контроль и управление доступом к информационным ресурсам в ЗОС «Феникс» реализованы на основе дискреционной и мандатной моделей безопасности. Кроме того, благодаря инвариантности взаимодействий и унификации операций УНИДИР, Феникс допускает применение любых моделей безопасности, основанных на контроле взаимодействий субъектов и объектов и управлении доступом с помощью атрибутов безопасности, без модификации микроядра и средств контроля доступа.

Пользователи в ЗОС «Феникс» представлены в виде строгой иерархии, на вершине которой находится администратор. Следование принципу повлекло за собой отказ от традиционного для ОС UNIX и MS Windows NT контроля доступа на основе степени доверенности приложений, когда полномочия пользователя определяются не столько назначенными ему правами доступа к объекту, сколько степенью привилегированности процесса, с помощью которого он осуществляет доступ к этому объекту.

С точки зрения безопасности все ресурсы ЗОС «Феникс», независимо от их типа, организованы в виде иерархии абстрактных объектов, доступ к которым унифицирован посредством операций УНИДИР, а все приложения независимо от их назначения считаются абстрактными субъектами, наследующими полномочия запустившего их пользователя. Поэтому алгоритмы контроля доступом универсальны для всех типов взаимодействий, а принятие решение о предоставлении/запрещении доступа зависит исключительно от значений индивидуальных атрибутов безопасности субъекта, объекта и вида запрашиваемой операции. Привилегированные приложения, с помощью которых пользователь может повышать свои полномочия, даже для выполнения отдельных операций, отсутствуют.

Уникальная системная архитектура ЗОС «Феникс» обеспечивает тотальный контроль всех взаимодействий в системе в соответствии с установленной политикой безопасности и предусматривает возможность расширения как состава защищаемых информационных ресурсов, так и методов осуществления доступа, что позволяет гарантировать соблюдение правил политики безопасности при выполнении любых операций доступа, независимо от способа его осуществления и природы информационных ресурсов.

7 Описание применения ЗОС «Феникс»

Пользователь может работать как за клавиатурой ПЭВМ, на котором установлена ЗОС «Феникс» (ПЭВМ «Феникс»), так и на отдельной рабочей станции, имеющей сетевое соединение с ПЭВМ «Феникс». В ЗОС «Феникс» информационные ресурсы разделяются, по своему размещению, на локальные - размещенные на ПЭВМ «Феникс», и удаленные - размещенные на других ПЭВМ, имеющих сетевое соединение с ПЭВМ «Феникс». На ПЭВМ «Феникс» защищаемыми информационными ресурсами (системными ресурсами) являются файлы и каталоги файловой системы.

Предоставляемые ЗОС «Феникс» средства прикладного программного интерфейса (API) и использование абстрактного понятия «ресурс» позволяют реализовать и обеспечить контроль и управление доступом к информационным ресурсам, определяемым пользователем (прикладные информационные ресурсы), размещенным на ПЭВМ «Феникс» или на других ПЭВМ ,имеющих сетевое соединение с ПЭВМ «Феникс». В этом случае ЗОС «Феникс» выполняет роль шлюза к прикладным (локальным и удаленным) ресурсам.

В зависимости от роли в процессе создания, обслуживания и потребления ресурсов в ЗОС «Феникс» существуют три основных типа компонентов: серверы, сервисы и приложения. Каждый компонент

представляется в виде одного или нескольких процессов в одном или нескольких экземплярах. Каждый тип характеризуется набором полномочий, условиями запуска и возможностью применения. С точки зрения функционирования системы, серверы являются компонентами, осуществляющими обработку запросов от других серверов, сервисов и приложений.

Сервер обслуживает совокупность однотипных, однозначно идентифицируемых ресурсов, которые отображаются в пространстве имен ЗОС «Феникс», и предоставляет к ним доступ посредством интерфейса УНИДИР. По выполняемым функциям серверы разделяются на:

- серверы аппаратных ресурсов,
- серверы информационных ресурсов,
- серверы объектов доступа,
- системные серверы.

Серверы аппаратных ресурсов обслуживают работу аппаратных устройств и реализуют к ним доступ по интерфейсу УНИДИР. Серверы информационных ресурсов обслуживают информационные ресурсы различного уровня, базирующиеся на аппаратных ресурсах. В отношении аппаратных ресурсов действует политика контроля и управления потреблением. Информационные ресурсы, которые попадают под действие политики безопасности, называются объектами. Серверы, которые регистрируют свои информационные ресурсы в системе контроля и управления доступом, называются серверами объектов. Только ресурсы, являющиеся объектами, могут быть доступны для сервисов и приложений, поэтому в отношении них действует контроль и управление доступом.

Отдельный класс серверов представляют системные серверы, или средства защиты, которые не поддерживают ресурсов, а осуществляют независимое функционирование. С точки зрения полномочий серверы, в отличие от приложений и сервисов, имеют доступ к части пространства ресурсов, ограниченной администратором, и не подчиняются общей политике безопасности. Сервисы и приложения имеют доступ только к пространству объектов, имеют полномочия пользователя, от имени которого исполняются, и контролируются политикой безопасности. Сервис отличается от приложения тем, что реализует некоторый прикладной протокол удаленного доступа.

Сервисы подразделяются на системные, выполняющиеся от имени системных идентификаторов (псевдопользователей), и прикладные, выполняющиеся от имени пользователя. Системные сервисы и серверы не могут быть запущены пользователем, а только администратором. Для доступа к системным информационным объектам локальному пользователю предоставлены команды и утилиты оболочки (командный интерпретатор текстовой консоли). Для доступа к системным информационным объектам удаленный пользователь может использовать стандартные (штатные) приложения FTP, HTTP или MS Network обмена. Пользователь может реализовать собственные приложения, обеспечивающие работу с системными или прикладными информационными объектами.

8 Перспективы развития ЗОС «Феникс»

В настоящее время разработчики работают над развитием ЗОС «Феникс» в следующих направлениях:

- повышение отказоустойчивости и обеспечение живучести ОС в экстремальных условиях эксплуатации, в т. ч. и при отказах аппаратуры;
- обеспечение поддержки 64-разрядных аппаратных средств (IA64, Alpha);
- портирование ЗОС «Феникс» на высокопроизводительные аппаратные платформы (Sparc, Alpha);
- создание на базе ЗОС «Феникс» распределенной сетевой среды хранения и обработки информации с унифицированным представлением ресурсов с помощью службы каталогов и контролем доступа на уровне домена;
- разработка инструментов создания прикладного программного обеспечения для ЗОС «Феникс» и системы программирования специального назначения на платформе ЗОС «Феникс»;

- разработка сетевого фильтра, встроенного в сетевую подсистему ЗОС «Феникс» с реализующего базовые функции межсетевого экрана и системы обнаружения вторжений;
- портирование в среду ЗОС «Феникс» системного и прикладного программного обеспечения с других Unix платформ (Linux, Solaris, BSD) и с платформы MS Windows (NT/2000).

Реализация системы управления доступом к информации в виде встраиваемых модулей аутентификации[13]

А. В. Галатенко, А. А. Наумов, А. Ф. Слепухин

В системах распределенного хранения и обработки информации важной компонентой является подсистема управления доступом. В докладе рассматриваются общие вопросы построения подсистемы аутентификации и авторизации, а также ее реализацию в рамках проекта PNIAM (Pluggable Non Interactive Authentication Modules, встраиваемые неинтерактивные модули аутентификации).

Современные системы управления доступом должны удовлетворять условиям целостности и гибкости. Целостность, в частности, обеспечивает то, что при наличии нескольких сервисов, обеспечивающих доступ к одной и той же информации (например, WWW и FTP), контроль прав доступа будет согласованным. Гибкость позволяет администратору оперативно реагировать на изменения в составе ресурсов и их пользователей.

В соответствии с изложенными требованиями разумно реализовывать систему управления доступом на основе принципов централизованности и модульности.

Под централизованностью понимается наличие единых, общих механизмов, не зависящих от конкретных приложений, которым требуется аутентификация и авторизация. Централизованность аутентификации и авторизации позволяет строить продуманные, гибкие и целостностные схемы управления доступом. Естественным способом реализовать принцип централизованности является выделение функций аутентификации, авторизации и протоколирования в отдельную библиотеку, предоставляющую абстрактный интерфейс приложениям.

Для достижения максимальной гибкости системы управления доступом независимые аутентификационные сервисы реализуются в виде отдельных динамически загружаемых модулей. Система динамической загрузки модулей позволяет заменять модули без перекомпиляции приложений и уменьшает потребление ресурсов.

В 1995 году эти три принципа были заложены в основу проекта PAM (Pluggable Authentication Modules, встраиваемые модули аутентификации). Преимущества централизованной, модульной, динамической схемы аутентификации были быстро оценены; PAM получил широкое распространение, став, например, частью популярной ОС RedHat. Однако по прошествии почти четырех лет стали очевидными и некоторые недостатки PAM.

- Разделение задач аутентификации и определения прав доступа и привилегий в некоторых случаях может иметь отрицательные последствия. Такое разделение оставляет теоретическую возможность несовместности процедур аутентификации и приобретения полномочий, или некорректных результатов второй.
- Неприспособленность PAM к выполнению задач аутентификации при отсутствии возможности интерактивного взаимодействия с пользователем. Существующая спецификация PAM разрешает модулям формировать запросы к пользователю без каких-либо ограничений на их количество и содержание. Это означает, что использование PAM в тех случаях, когда информационный обмен между сервером и клиентом строго фиксирован, затруднительно. Начало работ по решению этой проблемы было положено совместно с Andrew Morgan. Некоторые результаты этой работы можно найти в [1, 2].

[13]Работа частично выполнялась по программе фундаментальных научных исследований ОИТВС РАН «Оптимизация вычислительных архитектур под конкретные классы задач, информационная безопасность сетевых технологий».

- Для дальнейшего распространения PAM необходимо упрощение прикладного программного интерфейса.

PNIAM - это свободно распространяемая динамически загружаемая библиотека для ОС на базе ядра Linux, которая обеспечивает единообразное и настраиваемое выполнение процедур аутентификации пользователей, авторизации доступа к сервисам и сопутствующих процедур (протоколирование и др.). Основными сущностями сервисов PNIAM являются приложение (клиентская сторона) и библиотека со списком динамически загружаемых модулей (серверная сторона). Обмен информацией между клиентской и серверной сторонами осуществляется посредством структурированных поименованных элементов (в состав элемента входит поле «name», задающее тип информации (имеется список предопределенных типов, но по необходимости можно вводить и новые), и поле «data», определяющее собственно значение; в качестве примера может послужить элемент с name=USER и data=root). Таким образом, в отличие от PAM, в PNIAM обмен информацией стандартизован, что позволяет реализовывать неинтерактивные схемы аутентификации с тем же успехом, что и интерактивные.

Приложение может получить один из четырех сервисов:

- аутентификация (pniam_authenticate);
- авторизация (pniam_authorize);
- протоколирование (pniam_account_start - начало, pniam_account_end - конец);
- модификация пароля (pniam_change).

Библиотека PNIAM получает запрос на один из сервисов, и вызывает соответствующие динамически загруженные модули; каждый модуль принимает решение об успехе или неуспехе соответствующего сервиса и, возможно, запрашивает дополнительную информацию, после чего библиотека анализирует результаты работы модулей и принимает итоговое решение. Список используемых модулей и параметры их вызова определяются администратором системы с помощью конфигурационного файла.

К настоящему моменту реализована сама библиотека PNIAM и целый ряд PNIAM-модулей, предоставляющих широкий спектр сервисов аутентификации, авторизации, учета пользователей, смены аутентификационных токенов и проверки их валидности. Практически для каждого приложения, нуждающегося в описанных выше сервисах, имеется PNIAM-аналог. Ведутся работы по усилению методов проверки валидности, созданию PNIAM-версий популярных графических оболочек, а также по созданию централизованного распределенного сервера аутентификации на базе PNIAM. Более подробную информацию о PNIAM можно найти в [3].

Литература

[1] `http://kernel.org/pub/linux/libs/pam/`

[2] Morgan A. Pluggable Authentication Modules. IETF Internet Drafts, August 1998.

[3] `http://www.msu.ru/pniam/pniam.html`

Методы оценки эффективности управления и защиты транспортных соединений в высокоскоростных компьютерных сетях

Н. О. Вильчевский, В. С. Заборовский, В. Е. Клавдиев, Ю. А. Шеманин

Аннотация

Рассматриваются вопросы аналитического синтеза моделей процессов с целью оптимизации управления транспортными соединениями в высокоскоростных компьютерных сетях. С использованием аппарата однородных или вложенных цепей Маркова построены модели сетевых процессов и сформулированы два типа задач оптимизации обработки данных.

Ключевые слова: управление, оптимизация, параметры протокола, фрактальные распределения.

1 Введение

Глобальные компьютерные сети, модели которых в последнее время стали объектом интенсивных исследований, являются сложными техническими объектами, определяющими потенциальный уровень развития современной промышленности. Важной особенностью этих объектов является их сложная многослойная структура, объединяющая логические и физические подуровни с различными метриками и топологией в единую сетевую инфраструктуру. Проведенные исследования показывают, что за сложностью и многообразием сетевых процессов можно увидеть общие для всех открытых систем явления диссипации и упорядочивания. В конечном итоге именно они определяют динамику сетевого взаимодействия и приводят к возникновению устойчивых распределений и фрактальной структуре трафика. Подобные явления обычно связывают с действием процессов самоорганизации, свойства и модели которых изучает синергетика. Методология этой науки концентрируются вокруг того факта, что математическими моделями различных процессов в открытых системах оказались одни и те же уравнения. Исследование фрактальных сетевых процессов открывает новые перспективы для применения этой методологии с целью разработки алгоритмов оптимизации управления транспортными протоколами.

Цели управления в современных компьютерных сетях можно разделять на две категории — защита от несанкционированного использования данных и управление пропускной способностью транспортных соединений. Для этих, на первый взгляд различных целей, используются общие механизмы обработки пакетного трафика с помощью адаптивных процедур буферизации и обслуживания пакетов, многие из которых реализуются базе протоколов TCP/IP. Поэтому аналитический синтез алгоритмов управления TCP соединениями в среде высокоскоростных сетей коммутации пакетов, включая различные аспекты оптимизации их функционирования, является ключевой проблемой современной теории компьютерных телекоммуникаций. Ее решение, с одной стороны, требует учета сложного характера воздействия возмущающих факторов и явлений диссипативности, с которыми связывают снижением производительности устройств коммутации пакетов и потери пропускной способности, а с другой — критериев оптимальности, учитывающих различные аспекты производительности и надежности передачи пакетных данных.

В статье рассматривается модельная задача синтеза управления TCP соединением, которая позволяет выделить наиболее существенные характеристики процессов пакетной коммутации с учетом требований адаптации к состоянию сетевой среды и влияния возможных потерь. С использованием аппарата однородных или вложенных цепей Маркова построены модели сетевых процессов и сформулированы два типа задач оптимизации обработки данных.

2 Описание объекта и цели управления

Компьютерные сети можно рассматривать как распределенный объект управления, состояния которого подвержены стохастическим возмущениям. Применяемые транспортные протоколы адаптируются к текущему состоянию сети с помощью двух фаз в реализации каждого TCP соединения: фазе 1 — медленный старт и фазе 2 — защита от перегрузки. Адаптация задачи передачи пакетов данных через сеть обеспечивает за счет:

- надежность за счет повторной передачи неподтвержденных пакетов;
- занятие всей доступной полосы пропускания виртуального канала.

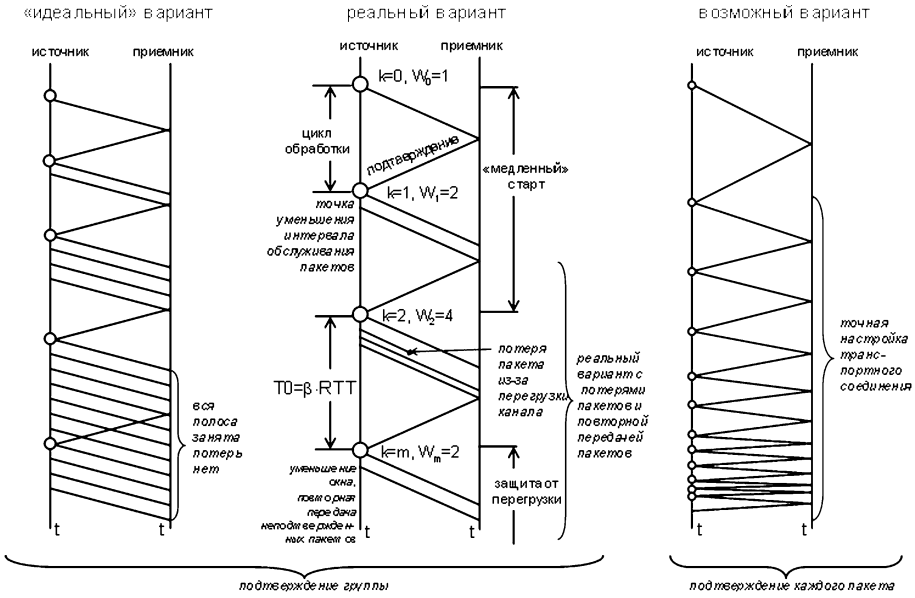


Рис. 1: Варианты реализации процедуры настройки TCP соединения.

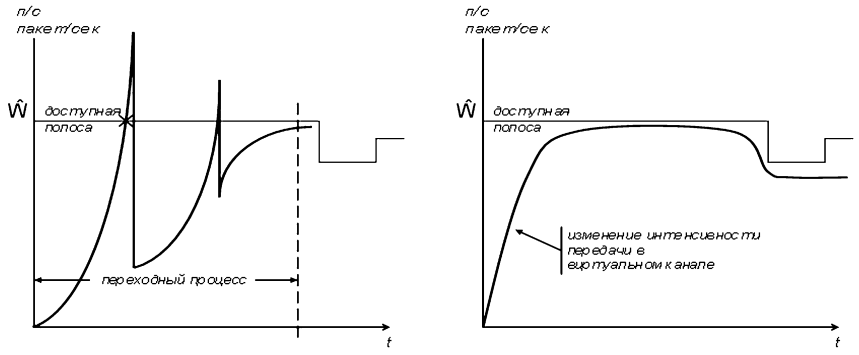


Рис. 2: Общий характер изменения интенсивности передачи пакетов в TCP канале.

Стандартное решение на базе протокола TCP основано на использовании механизма адаптации для выбора числа пакетов, отправленных в сеть до получения подтверждения:

$$W_{k+1} = 2W_k \quad \text{(доставка успешная)},$$
$$W_{k+1} = W_{k/2} \quad \text{(пакет не подтвержден за время T0, повторная передача)},$$
$$W_{k+1} = W_{k+1} \quad \text{(доставка успешная)},$$

где W_k — ширина окна, то есть число пакетов, отправленных в сеть до получения пакета подтверждения; k — момент регенерации алгоритма адаптации.

Использование такого механизма настройки протокола TCP на доступную полосу пропускания пакетов приводит к возникновению перегрузок, что сказывается на качестве функционирования информационных приложений, так как приводит к неизбежным потерям и резкому изменению интенсивности передачи пакетов (рис. 2а).

Поэтому настройка протоколов и изменения интенсивности передачи пакетов в виртуальных каналах, при которых характер переходных процессов соответствует ситуации, изображенной на рис. 2б, имеют важное значение для обеспечения высокого качества работы многих информационных приложений. Заметим, что эвристические методы улучшения работы транспортных протоколов хорошо

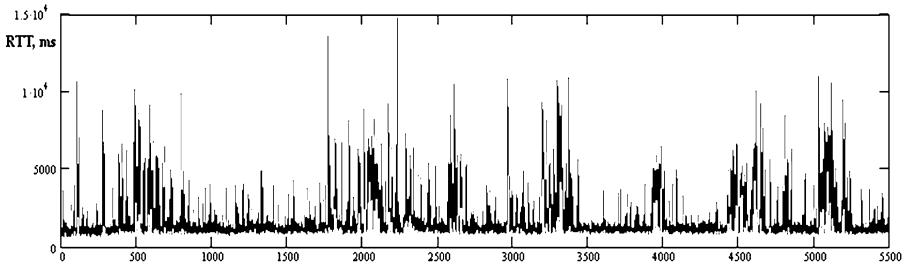


Рис. 3: Характер процессов в канале подтверждения успешного приема пакетов.

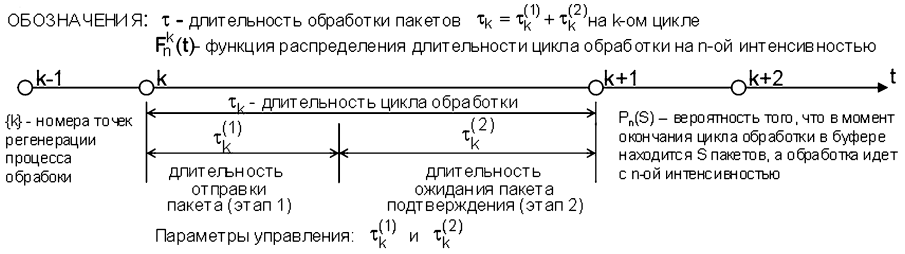


Рис. 4: Последовательность обработки пакетов в узлах TCP соединения.

известны, но их применение не позволяет изменить весьма сложный и непредсказуемый характер сетевых процессов и понять механизмы возникновения наблюдаемых явлений, в том числе, фрактальной структуры трафика (рис. 3).

Целью настоящей работы является получение модельных решений и анализ основных факторов, влияющих на пропускную способность транспортных соединений исходя их следующих предположений:

- Транспортное соединение имеет n уровней скорости обработки пакетов в узле-источнике.
- Выбор скорости обработки осуществляется на основе сигнала обратной связи с учетом функции плотности распределения длительности ожидания пакета подтверждения.
- Цикл обработки состоит из этапа подготовки к отправке и этапа ожидания подтверждения.

3 Метод решения

Рассмотрим систему обработки пакетов, последовательность этапов которой изображена на рис. 4.

Будем считать, что источник пакетов имеет буфер неограниченной вместимости, на вход которого поступает пуассоновский поток заявок (пакетов) интенсивностью λ. Обслуживание взятой из буфера заявки (пакета) осуществляется следующим образом. На непосредственную подготовку к отправке и саму отправку затрачивается случайное время t, после отправки пакета устройство обслуживания ожидает подтверждения факта приема пакета принимающем звеном в течении времени τ. Если за время τ не пришло этого подтверждения, то считается, что посланный пакет потерян, после чего длительность отправки пакетов замедляется и посылается дополнительный пакет, взамен утерянного. Если за время τ пришло подтверждение факта получения отправленного пакета и указание, что этот пакет испорчен, то длительность отправки пакетов не меняется, но взамен испорченного пакета посылается исправный. Если же за время τ пришло подтверждение факта получения пакета и указание на то, что этот пакет дошел без искажений, то длительность отправки нового пакета уменьшается. Будем считать, что система обработки пакетов может работать на двух возможных скоростях — минимальной (индексы, соответствующие этому уровню будем обозначать 0) и максимальной (соответствующий индекс равен 1).

Введем следующие обозначения: $F_n(t)$, $n = 0, 1$ — функция распределения суммарной (отправка пакета и ожидание ответа) длительности обработки последовательных пакетов при n-ой скорости; α_n

— вероятность того, что за время τ не поступит сообщение о приеме пакета принимающем звеном, а сама система работает на n-й скорости $(n = 0, 1)$; β — вероятность того, что искажения пакета в системе передачи не произошло; $q_n = (1 - \alpha_n)\beta$ — есть вероятность того, что за время τ пакет дошел без искажений, то есть вероятность того, что скорость отправки пакетов перейдет с 0-го на 1-й уровень, если работа осуществлялась на минимальной скорости, либо останется на максимальном уровне; $p_n = (1 - \alpha_n)(1 - \beta)$ $(n = 0, 1)$ — есть вероятность того, что пакет за время τ дошел до приемного звена, но был искажен, то есть скорость отправки пакетов не изменится, но будет послан новый, не испорченный пакет; $r_n = \alpha_n$ $(n = 0, 1)$ — есть вероятность того, что за время τ ответа о получении отправленного пакета принимающем звеном не получено, то есть скорость отправки пакетов перейдет с 1-го на 0-й уровень, если система работала на максимальном уровне, либо останется на минимальном уровне и в обоих случаях будет отправлен дополнительный пакет взамен утерянного; $p_n(s)$ $(n = 0, 1)$ — вероятность того, что в момент окончания цикла обработки отправляемого пакета в системе обслуживания находятся s пакетов и обработка следующего пакета будет осуществляться на n-ом уровне скорости передачи; $\pi_n = p_n(0)$ — вероятность того, что после окончания цикла обслуживания пакета в системе обслуживания отсутствуют пакеты, а вновь пришедший пакет будет обрабатываться на n-ом уровне скорости $(n = 0, 1)$.

Обозначим производящую функцию вероятностей $p_n(s)$, где $n = 0, 1$ как

$$G_n = \sum_{s=0}^{\infty} p_n(s) x^s.$$

Производящие функции для $n = 0, 1$ удовлетворяют следующей системе уравнений:

$$\begin{aligned} &(1 - (p_0 + r_0)L_0(x))G_0(x) - r_1 L_1(x) G_1(x) \\ &\qquad = (x - 1)((p_0 + r_0)L_0(x)\pi_0 + r_1 L_1(x)\pi_1), \\ &-\frac{q_0}{x} L_0(x) G_0(x) + \left(1 - \left(p_1 + \frac{q_1}{x}\right) L_1(x)\right) G_1(x) \\ &\qquad = (x - 1)\left(\frac{q_0}{x} L_0(x)\pi_0 + \left(p_1 + \frac{q_1}{x}\right) L_1(x)\pi_1\right) \end{aligned} \tag{1}$$

и условию нормировки, которое имеет вид

$$\sum_{n=0}^{1} \sum_{s=0}^{\infty} p_n(s) = G_0(1) + G_1(1) = 1. \tag{2}$$

В уравнении (1) введено следующее обозначение для преобразования Лапласа функции распределения длительности цикла обслуживания для n-ой скорости обработки пакетов

$$L_n(x) = \int_0^{\infty} e^{(x-1)\lambda t}\, dF_n(t),$$

где $n = 0, 1$.

Подставляя в сформулированные уравнения $x = 0$ и учитывая, что в силу введенных выше обозначений $G_n(0) = p_n(0) = \pi_n$, получаем равенство $\pi_0 = 0$. Это же вывод нетрудно получить из физических соображений. Действительно, находится в состоянии, когда скорость обработки пакетов минимальна система может лишь при условии, что отправленный на предыдущем шаге пакет либо не был получен принимающим звеном, либо был получен, но искажен в процессе передачи. В любом из этих вариантов, должен быть отправлен дополнительный пакет, поэтому в системе не могут отсутствовать пакеты и, следовательно, $p_1(0) = \pi_1 = 0$.

Учитывая это обстоятельство, сделаем в приведенной выше системе уравнений (1) следующую замену переменных:

$$G_0(x) = \mathcal{Q}_0(x), \quad G_1(x) = \mathcal{Q}_1(x) - (x - 1)\pi_1,$$

в результате получим:

$$\begin{aligned} &(1 - (p_0 + r_0)L_0(x))\mathcal{Q}_0(x) - r_1 L_1(x)\mathcal{Q}_1(x) = 0, \\ &-\frac{q_0}{x} L_0(x)\mathcal{Q}_0(x) + \left(1 - \left(p_1 + \frac{q_1}{x}\right) L_1(x)\right)\mathcal{Q}_1(x) = (x - 1)\pi_1. \end{aligned} \tag{3}$$

Решение (3) дается формулами:

$$\mathcal{Q}_0(x) = \frac{x-1}{\Delta(x)} r_1 L_1(x)\pi_1, \quad \mathcal{Q}_1(x) = \frac{x-1}{\Delta(x)}(1-(p_0+r_0)L_0(x))\pi_1,$$
$$\Delta(x) = (1-(p_0+r_0)L_0(x))\left(1-\left(p_1+\frac{q_1}{x}\right)L_1(x)\right) - \frac{q_0 r_1}{x} L_0(x)L_1(x).$$

Поэтому имеем

$$G_0(x) = \frac{x-1}{\Delta(x)} r_1 L_1(x)\pi_1, \quad G_1(x) = \frac{x-1}{\Delta(x)}(1-(p_0+r_0)L_0(x))\pi_1 - (x-1)\pi_1,$$
$$\text{где } \Delta(x) = (1-(p_0+r_0)L_0(x))\left(1-\left(p_1+\frac{q_1}{x}\right)L_1(x)\right) - \frac{q_0 r_1}{x} L_0(x)L_1(x). \tag{4}$$

Рассматривая эти выражения при $x = 1$ и раскрывая неопределенность по правилу Лопиталя, получим выражения

$$\Delta(1) = 0, \quad \Delta'(1) = q_0(q_1-\rho_1) + r_1(q_0-\rho_0),$$
$$G_0(1) = \frac{r_1}{q_0(q_1-\rho_1)+r_1(q_1-\rho_0)}\pi_1, \quad G_1(1) = \frac{q_0}{q_0(q_1-\rho_1)+r_1(q_1-\rho_0)}\pi_1.$$

4 Выбор критериев оптимизации

Полученные соотношения позволяют сформулировать критерии оптимизации управления TCP соединением. Учитывая условие нормировки, запишем

$$G_0(1) + G_1(1) = 1 \implies \pi_1 = \frac{\Delta'(1)}{r_1+q_0} = \frac{q_0(q_1-\rho_1)+r_1(q_1-\rho_0)}{r_1+q_0}. \tag{5}$$

С учетом этого соотношения получим выражение для производящих функций:

$$G_0(1) = \frac{r_1}{r_1+q_0}, \quad G_1(1) = \frac{q_0}{r_1+q_0}.$$

Очевидно, должны выполняться условия: $\pi_1 > 0 \implies q_0(q_1-\rho_1) + r_1(q_1-\rho_0) > 0$.

Так как $\pi_0 = 0$, то величине π_1 можно придать смысл вероятности того, что в системе отсутствуют пакеты, предназначенные для обработки. Следовательно, чем больше величина π_1, тем более эффективна работа системы. Поэтому с точки зрения управления представляется разумной постановка задачи оптимизации работы системы обработки, при которой величина π_1 достигает максимума.

Второй возможный критерий оптимальности работы системы связан с требованием минимизации величины средней очереди пакетов, ожидающих в очереди начала обслуживания. Величина средней очереди определяется следующим образом: $\overline{s} = \overline{s}_0 + \overline{s}_1 = G_0'(1) + G_1'(1)$. Дифференцируя (4), имеем

$$G_0(x) = \frac{x-1}{\Delta(x)} r_1 L_1(x)\pi_1, \quad G_1(x) = \frac{x-1}{\Delta(x)}(1-(p_0+r_0)L_0(x))\pi_1 - (x-1)\pi_1,$$
$$\text{где } \Delta(x) = (1-(p_0+r_0)L_0(x))\left(1-\left(p_1+\frac{q_1}{x}\right)L_1(x)\right) - \frac{q_0 r_1}{x} L_0(x)L_1(x),$$
$$G_0'(x) = \frac{\Delta(x)-(x-1)\Delta'(x)}{\Delta^2(x)} r_1 L_1(x)\pi_1 + \frac{(x-1)}{\Delta(x)} r_1 L_1'(x)\pi_1.$$

Переходя к пределу при $x = 1$, и раскрывая неопределенность по правилу Лопиталя, имеем:

$$G_0'(1) = -\frac{\Delta''(1)}{2(\Delta'(1))^2} r_1\pi_1 + \frac{1}{\Delta'(1)} r_1\rho_1\pi_1,$$

или, учитывая (5),

$$G_0'(1) = -\frac{\Delta''(1)}{2\Delta'(1)} \frac{r_1}{r_1+q_0} + \frac{\rho_1 r_1}{r_1+q_0}.$$

Аналогично получаем:

$$G_1'(1) = -\frac{\Delta''(1)}{2(\Delta'(1))^2} q_0\pi_1 + \frac{1}{\Delta'(1)}(-(1-q_0)\rho_0)\pi_1 - \pi_1,$$

и, учитывая (5), имеем

$$G_1'(1) = -\frac{\Delta''(1)}{2\Delta'(1)}\frac{q_0}{r_1+q_0} - (1-q_0)\frac{\rho_0}{r_1+q_0} - \frac{q_0(q_1-\rho_1)+r_1(q_0-\rho_0)}{r_1+q_0}.$$

Здесь

$$\begin{aligned}\Delta''(1) = 2(&\rho_0\rho_1 - \rho_0\rho_1 r_1 - \rho_0\rho_1 q_0 + r_1 q_0\rho_1 + r_1 q_0\rho_0 \\ &- r_1 q_0 + q_0 q_1\rho_1 + q_0 q_1\rho_0 - q_0 q_1 - \rho_0 q_1) - q_0 V_1 - r_1 V_0.\end{aligned}$$

Причем $V_i = \frac{d^2 L_i(x)}{dx^2}\Big|_{x=1} = \lambda^2 \int_0^\infty t^2\, dF_i(t) = \rho_i^2 + \lambda^2\sigma_i^2$, $(i = 0, 1)$, где σ_i^2 — дисперсия цикла времени обработки пакетов.

В результате, получаем

$$\overline{s} = -\frac{\Delta''(1)}{2\Delta'(1)} - \frac{\rho_0 - ((\rho_0+\rho_1)(q_0+r_1) - q_0(q_1+r_1))}{r_1+q_0}. \tag{6}$$

В полученных формулах введены следующие обозначения: π — вероятность отсутствия пакетов в буфере, r_i — вероятность того, что подтверждения о получении пакета не поступило, $q_i = \alpha(1-r_i)$ — вероятность получить подтверждение того, что отправленный пакет дошел без искажений, α — вероятность порчи пакета, $\rho_i = \lambda T_i$ — коэффициент загрузки системы, T_i — среднее время цикла обработки пакета, $F_i(u)$ — функция распределения длительности цикла обслуживания, а $L_i(x) = \int_0^\infty e^{(x-1)\lambda u}\, dF_i(u)$, $(i = 1, 2)$. В качестве оценки величины r_i примем выражение:

$$r_i = \int_0^\infty (1 - \mathcal{Q}(\tau))\, dF_i^{(2)}(\tau), \quad i = 1, 2,$$

где $\mathcal{Q}(\tau)$ — функция распределения времени получения подтверждения о приеме пакета, $F_i^{(2)}$ — функция распределения длительности второго этапа цикла обработки пакетов. При этом, если все времена обслуживания пакетов распределены по экспоненциальному закону, то $r_i = \frac{t}{t+t_i^{(2)}}$, где t — среднее время получения подтверждения, $t_i^{(2)}$ — средняя длительность второго этапа цикла.

Полученные соотношения можно обобщить на случай $n > 2$ уровней интенсивностей обслуживания заявок. Будем рассматривать состояния системы в моменты окончания обслуживания заявок, то есть только в моменты, когда процесс является Марковским. Обозначим q_k вероятность того, в случае, когда система работает в k-ом режиме, пакет дошел без искажений, p_k — вероятность того, что он был поврежден, r_k — вероятность того, что информация о получении пакета принимающим узлом не поступила.

Введем вероятности $P\left(\begin{smallmatrix} n & k \\ s & l \end{smallmatrix}\right)$ того, что в момент окончания обслуживания пакета в системе находится n пакетов и обслуживание осуществляется в k-м режиме, в состояние, когда в момент окончания обслуживания следующего пакета в системе находится s пакетов и обслуживание осуществляется в l-м режиме. Рассмотрим $\pi(s,k)$ — стационарные вероятности того, что в системе находится s пакетов и обслуживание осуществляется в k-м режиме. Эти вероятности удовлетворяют следующим

уравнениям:

$$\begin{aligned}
\pi(s,0) &= P\begin{pmatrix}0 & 0\\ s & 0\end{pmatrix}\pi(0,0) + P\begin{pmatrix}0 & 1\\ s & 0\end{pmatrix}\pi(0,1)\\
&\quad + \sum_{n=1}^{s} P\begin{pmatrix}n & 0\\ s & 0\end{pmatrix}\pi(n,0) + \sum_{n=0}^{s} P\begin{pmatrix}n & 1\\ s & 0\end{pmatrix}\pi(0,1),\\
\pi(s,k) &= P\begin{pmatrix}0 & k-1\\ s & k\end{pmatrix}\pi(0,k-1) + P\begin{pmatrix}0 & k\\ s & k\end{pmatrix}\pi(0,k)\\
&\quad + P\begin{pmatrix}0 & k+1\\ s & k\end{pmatrix}\pi(0,k+1) + \sum_{n=1}^{s+1} P\begin{pmatrix}n & k-1\\ s & k\end{pmatrix}\pi(n,k-1)\\
&\quad + \sum_{n=1}^{s} P\begin{pmatrix}n & k\\ s & k\end{pmatrix}\pi(n,k) + \sum_{n=1}^{s} P\begin{pmatrix}n & k+1\\ s & k\end{pmatrix}\pi(n,k+1),\\
\pi(s,N) &= P\begin{pmatrix}0 & N-1\\ s & N\end{pmatrix}\pi(0,N-1) + P\begin{pmatrix}0 & N\\ s & N\end{pmatrix}\pi(0,N)\\
&\quad + \sum_{n=1}^{s+1} P\begin{pmatrix}n & N-1\\ s & N\end{pmatrix}\pi(n,N-1) + \sum_{n=1}^{s} P\begin{pmatrix}n & N\\ s & N\end{pmatrix}\pi(n,N).
\end{aligned}$$

Следуя рассмотренной ранее методики вывода уравнений состояния, введем производящие функции для вероятностей

$$G_k(x) = \sum_{s=0}^{\infty} \pi(s,k)x^s.$$

Используя стандартные обозначения для преобразования Лапласа функции распределения длительности цикла обслуживания

$$L_k(x) = \int_0^{\infty} e^{(x-1)\lambda t}\, dF_k(t),$$

получим:

$$\begin{aligned}
G_0(x) &= (p_0+r_0)\sum_{s=1}^{\infty}\int_0^{\infty}\frac{(\lambda t)^{s-1}}{(s-1)!}x^s e^{-\lambda t}\, dF_0(t)\pi(0,0)\\
&\quad + r_0\sum_{s=1}^{\infty}\int_0^{\infty}\frac{(\lambda t)^{s-1}}{(s-1)!}x^s e^{-\lambda t}\, dF_0(t)\pi(0,1)\\
&\quad + (p_1+r_1)\sum_{s=1}^{\infty}\sum_{n=1}^{s}\int_0^{\infty}\frac{(\lambda t)^{s-n}}{(s-n)!}x^s e^{-\lambda t}\, dF_0(t)\pi(n,0)\\
&\quad + r_0\sum_{s=1}^{\infty}\sum_{n=1}^{s}\int_0^{\infty}\frac{(\lambda t)^{s-n}}{(s-n)!}x^s e^{-\lambda t}\, dF_0(t)\pi(n,1).
\end{aligned}$$

После преобразований имеем:

$$\begin{aligned}
G_0(x) &= (p_0+r_0)xL_0(x)\pi(0,0) + r_1xL_1(x)\pi(0,1) +\\
&\quad + (p_0+r_0)xL_0(x)(G_0(x)-\pi(0,0)) + r_1xL_1(x)(G_1(x)-\pi(0,1)),
\end{aligned}$$

что позволяет окончательно получить выражение для случая $n=0$

$$\begin{aligned}
G_0(x) &= (p_0+r_0)L_0(x)G_0(x) + r_1L_1(x)G_1(x) +\\
&\quad + (x-1)((p_0+r_0)L_0(x)\pi(0,0) + r_1L_1(x)\pi(0,1)).
\end{aligned}$$

Аналогично для $n = k$, получаем выражение:

$$G_k(x) = \frac{q_{k-1}L_{k-1}(x)}{x}G_{k-1}(x) + p_kL_k(x)G_k(x) + r_{k+1}L_{k+1}G_{k+1}(x)$$
$$+ (x-1)\left(\frac{q_{k-1}L_{k-1}(x)}{x}\pi(0,k-1) + p_kL_k(x)\pi(0,k) + r_{k+1}L_{k+1}(x)\pi(0,k+1)\right),$$
$$G_N(x) = \frac{q_{N-1}L_{N-1}(x)}{x}G_{N-1}(x) + \left(\frac{q_N}{x} + p_n\right)L_N(x)G_N(x)$$
$$+ (x-1)\left(\frac{q_{N-1}L_{N-1}(x)}{x}\pi(0,N-1) + \left(\frac{q_N}{x} + p_n\right)L_N(x)\pi(0,N)\right).$$

Полагая в этих уравнениях $x = 1$ и учитывая, что $L_k(1) = 1$ и $p_k + r_k + q_k = 1$, запишем:

$$\begin{aligned} &q_0G_0(1) - r_1G_1 = 0,\\ -&q_{k-1}G_{k-1}(1) + (r_k + q_k)G_k(1) - r_{k+1}G_{k+1}(1) = 0,\\ -&q_{N-1}G_{N-1} + r_NG_N(1) = 0. \end{aligned}$$

Откуда имеем

$$G_k(1) = \frac{\prod_1^k r_s}{\prod_0^{k-1} q_s}G_0(1).$$

Так как $\sum_0^N G_k(1) = 1$, то получаем

$$G_0(1) = \frac{1}{1 + \sum_{k=1}^N \frac{\prod_1^k r_s}{\prod_0^{k-1} q_s}}.$$

5 Заключение

Сформулированные выражения для критериев (5) и (6) позволяют осуществить синтез транспортных протоколов, обеспечивающих защиту виртуальных каналов от перегрузок с учетом требований по оптимизации их режимов работы. Впервые в практике разработки TCP протоколов предложен подход, основанный на аналитическом решении задачи управления интенсивностью обслуживания. При этом могут быть учтены реальные, в том числе фрактальные, статистические характеристики сетевой среды с помощью задания соответствующего распределения $Q(\tau)$. Для практической реализации данного подхода особое значение имеет тот факт, что вид функции распределения может уточняться на основе экспериментальных данных в процессе функционирования сети.

Приложение. Расчет вероятностей перехода

$$
\begin{aligned}
P\begin{pmatrix}0 & 0\\ s & 0\end{pmatrix} &= p_0\int_0^\infty \frac{(\lambda t)^{s-1}}{(s-1)!}e^{-\lambda t}\,dF_0 + r_0\int_0^\infty \frac{(\lambda t)^{s-1}}{(s-1)!}e^{-\lambda t}\,dF_0,\\
P\begin{pmatrix}0 & 0\\ s & 1\end{pmatrix} &= q_0\int_0^\infty \frac{(\lambda t)^{s}}{s!}e^{-\lambda t}\,dF_0,\\
P\begin{pmatrix}0 & k\\ s & k-1\end{pmatrix} &= r_k\int_0^\infty \frac{(\lambda t)^{s-1}}{(s-1)!}e^{-\lambda t}\,dF_k,\\
P\begin{pmatrix}0 & k\\ s & k\end{pmatrix} &= p_k\int_0^\infty \frac{(\lambda t)^{s-1}}{(s-1)!}e^{-\lambda t}\,dF_k,\\
P\begin{pmatrix}0 & k\\ s & k+1\end{pmatrix} &= q_k\int_0^\infty \frac{(\lambda t)^{s}}{s!}e^{-\lambda t}\,dF_k,\\
P\begin{pmatrix}0 & N\\ s & N-1\end{pmatrix} &= p_N\int_0^\infty \frac{(\lambda t)^{s-1}}{(s-1)!}e^{-\lambda t}\,dF_N,\\
P\begin{pmatrix}0 & N\\ s & N\end{pmatrix} &= p_N\int_0^\infty \frac{(\lambda t)^{s-1}}{(s-1)!}e^{-\lambda t}\,dF_N + q_N\int_0^\infty \frac{(\lambda t)^{s}}{s!}e^{-\lambda t}\,dF_N,\\
P\begin{pmatrix}n & k\\ s & k-1\end{pmatrix} &= r_k\int_0^\infty \frac{(\lambda t)^{s-n}}{(s-n)!}e^{-\lambda t}\,dF_k,\\
P\begin{pmatrix}n & k\\ s & k\end{pmatrix} &= p_k\int_0^\infty \frac{(\lambda t)^{s-n}}{(s-n)!}e^{-\lambda t}\,dF_k,\\
P\begin{pmatrix}n & k\\ s & k+1\end{pmatrix} &= q_k\int_0^\infty \frac{(\lambda t)^{s-n+1}}{(s-n+1)!}e^{-\lambda t}\,dF_k,\\
P\begin{pmatrix}n & N\\ s & N\end{pmatrix} &= p_N\int_0^\infty \frac{(\lambda t)^{s-n}}{(s-n)!}e^{-\lambda t}\,dF_N + q_N\int_0^\infty \frac{(\lambda t)^{s-n+1}}{(s-n+1)!}e^{-\lambda t}\,dF_N.
\end{aligned}
$$

Анализ нормативно-методической базы для создания VPN на основе семейства протоколов IPSec с использованием автоматического управления ключом и инфраструктуры открытого ключа

В. А. Сухомлин, О. Р. Лапонина

В работе сделан анализ современного состояния нормативно-методической базы для создания Virtual Private Network (VPN) на основе семейства протоколов IPSec. Основное внимание уделено анализу возможности использования в протоколах IPSec распределения ключей с использованием криптографии с открытым ключом и поддерживающей ее инфраструктуры.

В работе сделан анализ пакета стандартов, используемых для построения профилей VPN-технологий. Согласованность стандартов является важнейшим фактором при реализации открытых систем и обеспечивает высокую степень интероперабельности продуктов, реализующих VPN-технологии. Целью подобного анализа является формирование основы для построения профилей, которые могли бы являться важнейшим инструментом реализации открытых систем.

1 IPSec

В настоящее время большой интерес представляет развертывание VPN, основанных на продуктах семейства протоколов IPSec. Цель этого семейства протоколов состоит в том, чтобы обеспечить различные сервисы безопасности трафика на уровне IP в окружении как IPv4, так и IPv6. Набор сервисов безопасности включает управление доступом, целостность соединения, первичную аутентификацию данных, защиту против атак повтора (replays), т. е. целостность последовательности, конфиденциальность как прикладных данных, так и всего трафика (с использованием шифрования). Эти сервисы предоставляются на уровне IP, обеспечивая защиту для IP и/или протоколов более высокого уровня. Рассмотрим основные компоненты архитектуры безопасности IPsec:

1. Протоколы безопасности - Authentication Header (AH) и Encapsulating Security Payload (ESP).
2. Безопасные Ассоциации - Security Association (SA).
3. Управление ключом - ручное и автоматическое, основанное на протоколе IKE версии 1 и 2.

Authentication Header (AH) обеспечивает целостность соединения, аутентфикацию исходных данных и дополнительно может предоставлять anti-replay сервис.

Encapsulating Security Payload (ESP) протокол может обеспечивать конфиденциальность (шифрование) и целостность соединения, аутентификацию исходных данных и дополнительно может предоставлять anti-replay сервис. (Один или другой набор этих сервисов должен быть применен всякий раз, когда используется ESP).

Каждый протокол поддерживает два режима использования: транспортный режим и режим туннелирования. В транспортном режиме протоколы обеспечивают защиту главным образом для протоколов более высокого уровня; в режиме туннелирования протоколы применяются для туннелирования IP пакетов.

Security Association (SA) определяет параметры соединения, с помощью которого обеспечивается возможность функционирования протоколов безопасности. SA однозначно определяется тройкой, состоящей из Security Paramener Index (SPI), IP Destination Address и идентификатора протокола безопасности (AH или ESP).

IPSec требует строгую аутентификацию для предотвращения активных атак. Основная цель строгой аутентификации в IPSec состоит в том, чтобы гарантировать целостность обменов ключа. Хорошо известно, что неаутентифицированный Диффи-Хеллман уязвим для простой активной атаки. Установление разделяемого секрета вручную не является достаточно хорошо масштабируемым решением. Следовательно необходим протокол для динамического установления данного состояния - IKE второй версии.

IKE выполняет взаимную аутентификацию между двумя участниками и устанавливает безопасную ассоциацию IKE, включающую разделяемую секретную информацию, которая может быть использована для эффективного установления SAs для ESP и/или AH, и множество криптографических алгоритмов, используемых для защиты SAs. Будем обозначать IKE SA как IKE-SA. SAs для ESP и/или AH, которые получены из данного IKE-SA, будем обозначать CHILD-SA.

Все IKE взаимодействия состоят из пар сообщений: запрос и ответ. Такая пара называется «обмен» («exchange»). Будем называть первые сообщения, устанавливающие IKE-SA обменами IKE-SA-INIT и IKE-AUTH и последующие обмены IKE обменами CREATE-CHILD-SA и INFORMATIONAL. В общем случае существует единственный IKE-SA-INIT обмен и единственный IKE-AUTH обмен (всего четыре сообщения) для установления IKE-SA и первого CHILD-SA. В исключительных случаях может быть более одного обмена для каждого варианта. Во всех случаях все IKE-SA-INIT обмены должны завершиться до начала любого другого типа обмена. Все IKE-AUTH обмены должны завершиться, следовательно произойти любое количество CREATE-CHILD-SA и INFORMATIONAL обменов в любой последовательности. В некоторых сценариях необходим только единственный CHILD-SA между конечными точками IPSec и, следовательно, не будет дополнительных обменов. Последующие обмены могут использоваться для установления дополнительных CHILD-SAs между теми же самыми аутентифицированными парами конечных точек и выполнения соотвествующих функций.

Поток сообщений IKE всегда состоит из запроса, за которым следует ответ. На запрашивающей стороне лежит ответственность за гарантирование надежности. Если ответ не получен в определенный интервал времени, запрашивающая сторона должна повторить запрос (или разорвать соединение). В первом запросе/ответе сессии IKE ведутся переговоры о параметрах безопасности для

IKE-SA, посылаются nonces и посылаются значения Диффи-Хеллмана. Будем называть начальный обмен IKE-SA-INIT (запрос и ответ).

Второй запрос/ответ, который будем называть IKE-AUTH, передает идентификации, доказывает знание секретов, относящихся к обоим идентификациям, и устанавливает SA для первого (и часто единственного) AH и/или ESP CHILD-SA.

Следующими типами обменов являются CREATE-CHILD-SA (который создает CHILD-SA) и INFORMATIONAL (который удаляет SA, сообщает об ошибочных условиях и выполняет некоторую другую работу). Каждый запрос требует ответа. INFORMATIONAL запрос без содержимого обычно используется для проверки жизнеспособности. Данная последовательность обменов не может быть использована до тех пор, пока не завершаться начальные обмены.

2 PKI

Для автоматической аутентификации необходимо использовать криптографию с открытым ключом, что в свою очередь требует наличия инфраструктуры открытого ключа (PKI).

ITU-T X.509 и ISO/IEC 9594-8 определили стандартный формат PKC (Public Key Certificate - сертификат открытого ключа), впервые опубликовав его в 1998 году как часть рекомендаций Директории X.500. Формат PKC в стандарте 1998 года называется форматом версии 1 (v1).

При пересмотре в 1993 году X.500 были добавлены поля subjectUniqueIdentifier и issuerUniqueIdentifier, в результате чего появился формат версии 2. Эти два поля могут быть использованы для поддержки управления доступом в директорию.

RFCs для PEM, опубликованные в 1993 году, включали спецификации для инфраструктуры открытого ключа, основываясь на X.509 v1 сертификатах открытого ключа. Результаты, полученные при попытки развернуть PEM, привели к тому, что стало ясно, что форматы сертификатов открытого ключа v1 и v2 имеют недостатки. Самое важное, что необходимо больше полей. В ответ на эти новые требования был разработан стандарт версии 3 (v3). Формат v3 расширяет формат v2 добавлением заготовок для дополнительных полей расширения. Конкретные типы полей расширения могут быть специфицированы в стандартах или могут быть определены и зарегистрированы любой организацией или сообществом. В июне 1996 года стандартизация базового формата v3 была завершена.

Были также разработаны стандартные расширения для использования в поле расширений v3. Эти расширения могут охватывать такие данные как дополнительную информацию об идентификации субъекта, информацию атрибута ключа, информацию политики и ограничения сертификационного пути. Однако стандартные расширения ISO/IEC/ITU и ANSI X9 очень широки для их применимости. Для разработки интероперабельных реализаций систем X.509 v3 для использования в интернет необходимо специфицировать профиль для использования расширений X.509 v3, предназначеннный для интернет. Одной из целей PKIX является спецификация профиля для приложений интернет, электронной почты, IPSec и т. д.

Цель PKI состоит в обеспечении детерминированной, автоматической идентификации, аутентификации, управлении доступом и функций авторизации. Поддержка этих сервисов определяет атрибуты, содержащиеся в сертификате, а также вспомогательную управляющую информацию в сертификате, такую как данные политики и ограничения сертификационного пути.

PKI подразумевает определение

1. Форматов сертификатов открытого ключа.
2. Алгоритмов проверки сертификатов.
3. Протоколов взаимодействия всех участников PKI.

Стандартами определены профили сертификата X.509 v3 и списка отмененных сертификатов (CRL) X.509 v2, в которых детально описан формат сертификата X.509 v3 в соответствии с семантиками способов именования в интернете, описаны стандартные расширения сертификата и определены два расширения, специфичных для интернета, описан набор обязательных расширений сертификата, описан в деталях формат CRL X.509 v2, определены обязательные расширения, описан алгоритм проверки действительности сертификационного пути X.509.

2.1 Функции PKI

2.1.1 Регистрация

Это процесс, при котором субъект впервые сообщает о себе CA (непосредственно или через RA) до того как CA выпустит PKC или PKCs для данного субъекта. Регистрация включает предоставление имени (например, общее имя, полностью определенное доменное имя, IP адрес) и другие атрибуты, размещаемые в PKC, далее CA (возможно с помощью RA) проверяет в соответствии со своим утверждением о сертификационной практики, что имя и другие атрибуты корректны.

2.1.2 Инициализация

Инициализация требуется тогда, когда субъект (например, пользователь или клиентская система) получает значения, необходимые для начала взаимодействия с PKI. Например, инициализация может включать предоставление клиентской системе открытого ключа или PKC CA или создание клиентской системой своей собственной пары открытый-закрытый ключ.

2.1.3 Сертификация

Это процесс, при котором CA выпускает PKC для открытого ключа субъекта и возвращает этот PKC субъекту или передает этот PKC в репозиторий.

2.1.4 Восстановление пары ключей

В некоторых реализациях локальная политика требует, чтобы ключи обмена ключа или шифрования были «архивированы» или восстанавливаемы в случае, если ключ потерян и необходим доступ к ранее зашифрованной информации. Возможны случаи, что ключ обмена закрытого ключа хранится в аппаратном токене, который может быть потерян или взломан, или файл закрытого ключа защищен паролем, который может быть забыт. Часто считается, что компания может читать зашифрованную почту конкретного сотрудника, когда данный сотрудник не доступен, например более не работает в компании.

В этих случаях для закрытого ключа пользователя может быть сделан back up специальной системой или самим CA. Если пользователю необходимо восстановить этот материал ключа, PKI должен предоставить систему, которая позволяет восстановление без неприемлемого риска компрометации закрытого ключа.

2.1.5 Создание ключа

В зависимости от политики CA пара закрытый-открытый ключ может создаваться либо пользователем в своем локальном окружении, либо создаваться CA. Во втором случае материал ключа может предоставляться пользователю в виде зашифрованного файла или физического токена (например, смарт-карты).

2.1.6 Изменение ключа

Все пары ключей необходимо регулярно изменять (например, заменять новой парой ключей) и выпускать новые PKCs. Это может произойти в двух случаях: нормально, когда ключ максимально исчерпал свое время жизни; исключительно, когда ключ был компрометирован и должен быть заменен.

2.1.7 Истечение срока ключа

В обычном случае PKI должна обеспечивать возможность плавного перехода от PKC с существующим ключом к PKC с новым ключом. Это особенно важно, когда изменяемый ключ является ключом CA. Пользователи должны знать от истечении ключа в определенное время; PKI, функционируя совместно с использующими PKI приложениями, следует предусматривать функционирование до и после изменения соответствующего ключа. Существует несколько способов сделать это; см [CMP] в качестве одного из примеров.

2.1.8 Компрометация ключа

В случае компрометации ключа замена не может быть «плавной», так как это незапланированная смена PKCs и ключей; пользователи заранее об этом не знают. Тем не менее PKI должно поддерживать возможность объявления, что предыдущий PKC более недействительный и не должен использоваться и выпустить уведомление о действительности и доступности нового PKC.

Замечание: компрометация закрытого ключа, связанного с корневым CA, является катастрофической для пользователей, доверяющих данному корневому CA. Если закрытый ключ корневого CA компрометирован, данный PKC CA должен быть отменен и все подчиненные PKCs также должны быть отменены. До того времени как корневой CA выпустит новый PKC и PKCs пользователей, доверяющих ему, эти пользователи не могут воспользоваться системой, так как не существует способов создания действительного сертификационного пути к доверенному узлу.

Затем пользователи должны быть уведомлена внешними механизмами об изменении ключей CAs. Если старый ключ компрометирован, любое сообщении об «изменении», требующее от подчиненных переключения на новый ключ, может прийти от атакующего, обладающего старым ключом, и может указывать на новый открытый ключ, для которого атакующий уже имеет закрытый ключ. Следует предвидеть это событие и предоставить ключ корневого CA всем доверяющим группам некоторым безопасным, внешним механизмом.

Дополнительно как только корневой CA получит новый ключ, ему будет необходимо перевыпустить PKCs, подписав их новым ключом, для всех подчиненных пользователей, т. к. их текущий PKC подписан отмененным ключом.

2.1.9 Кросс-сертификация

Сертификат CA является сертификатом в иерархии, который не является ни самоподписанным сертификатом, ни сертификатом конечного участника. [2459bis] не делает различия между сертификатом CA и кросс-сертификатом, так как он определяет кросс-сертификат как «сертификат, выпущенный одним CA для другого CA». Некоторые члены WG считают, что кросс-сертификат является специальным типом сертификата CA. Кросс-сертификат выпущен CA под одним Top CA для другого CA под другим Top CA. CAs в одной и той же иерархии содержат в части своего имени Top CA или CAs под Top CA. Когда выпускается кросс-сертификат, не существует взаимосвязи между именами CAs.

Обычно кросс-сертификат используется для того, чтобы позволить клиентским системам или конечным учатникам в одном административном домене безопасно взаимодействовать с клиентскими системами или конечными пользователями в другом административном домене. Использование кросс-сертификата, выпущенного CA-1 для CA-2, позволяет пользователю Алисе, которая доверяет CA-1, принимать PKC, используемый Бобом, сертификат которого был выпущен CA-2. Если требуется, кросс-сертификаты могут также быть выпущены одним CA для другого CA в том же самом административном домене.

Кросс-сертификаты могут быть выпущены как только в одном направлении, так и в обоих направлениях между двумя CAs. Это означает, что только потому, что CA-1 выпустил кросс-сертификат для CA-2, CA-2 еще не должен выпускать кросс-сертификат для CA-1.

2.1.10 Отмена

Когда PKC выпущен, считается, что он будет использоваться в течении всего периода действительности. Однако могут произойти различные события, которые приведут к тому, что PKC станет недействительным до истечения периода действительности. Такие обстоятельства включают изменение имени, изменение связи между субъектом и CA (например, сотрудник прекращает работу в организации), а также компрометация или предположение компрометации соответствующего закрытого ключа. При таких обстоятельствах CA необходимо отменить PKC.

X.509 определяет один метод отмены PKC. Этот метод состоит в том, что каждый CA периодически выпускает подписанную структуру данных, называемую CRL. CRL является списком, который содержит ссылки на отмененные PKCs. Этот список содержит дату выпуска, он подписывается CA и делается свободно доступным в открытом репозитории. Каждый отмененный PKC идентифицируется в CRL своим серийным номером. Когда использующая сертификаты система получает PKC, эта система не только проверяет подпись и действительность PKC, но также также запрашивает соответствующий CRL и проверяет, что серийный номер PKC не находится в данном CRL. Значение

«соответствующий» может зависеть от локальной политики, но обычно означает самый последний выпущенный CRL. CA выпускает новый CRL на регулярной периодической основе (например, ежечасно, ежедневно или еженедельно). CAs могут также выпускать CRLs непериодически. Например, если компрометирован важный ключ, CA может выпустить новый CRL для отправки оповещения об этом факте, даже если следующий CRL не должен быть выпущен в данное время. (Проблема непериодических экземпляров CRL состоит в том, что конечные участники могут не узнать, что был выпущен новый CRL, и поэтому не запросить его из репозитория).

Запись добавляется в CRL как часть следующего оповещения об отмене. Запись может быть удалена из CRL после появления в одном из регулярно выпускаемых CRL. Если отмененный PKC не хранится в CRL далее периода действительности, то возможна ситуация, что отмененный PKC никогда не появится в CRL.

Преимущество метода отмены с помощью CRL состоит в том, что CRLs могут распространяться тем же способом, что и сами PKCs, а именно по недоверяемым коммуникациям и серверным системам.

Одно ограничение метода отмены с помощью CRL при использовании недоверенных коммуникаций и серверов состоит в том, что временная точность ограничена периодом выпуска CRL.

Как и в случае с форматом X.509 v3 PKC для достижения интероперабельности необходимо профилирование формата X.509 v2 CRL. Это сделано как часть профиля PKI [FORMAT]. Однако PKIX не требует, чтобы CAs выпускали CRLs. В некоторых окружениях могут применяться on-line методы уведомления об отмене в качестве альтернативы X.509 CRL. PKIX определяет несколько протоколов, которые поддерживают on-line проверку. OCSP, DVCS и SCVP поддерживают on-line проверку статуса PKCs.

On-line проверка отмены может быть важна для уменьшения задержки между сообщением об отмене и распространением информации доверяющим группам. После того как CA принимает сообщение об отмене как аутентичное и действительное, любой запрос к on-line сервису будет корректно отражать действительность PKC. Однако эти методы накладывают новые требования безопасности; проверяющий действительность PKC должен доверять on-line сервису проверки действительности, в то время как репозиторий не обязательно должен быть доверяемым.

2.1.11 Распространение уведомления о сертификате и отмене, публикование

Как говорилось выше, PKI ответственен за распространение PKCs и уведомлений об отмене PKCs (как в форме CRL, так и в какой-то другой форме). «Распространение» PKCs включает передачу PKC своим собственникам, а также может включать публикование PKC в репозитории. «Распространение» уведомления об отмене включает помещение CRLs в репозиторий, пересылку их конечным участникам или передачу их при on-line запросах.

2.2 Протоколы PKI

Параллельно с определением профиля сертификата в организациях, занимающихся разработкой стандартов, проектировались протоколы, необходимые для управления информацией, относящейся к PKI. Первым был разработан протокол управления сертификатом (CMP). Он определяет последовательность сообщений для инициализации, сертификации, изменения и отмены участников PKI. Проект синтаксиса запроса сертификата (CRS) был разработан в SMIME WG, которая использовала PKCS-10 в качестве формата сообщения запроса сертификата. Был также разработан проект формата сообщения запроса сертификата (CRMF), но в PKIX WG. Он определяет простой протокол запроса, который является подмножеством обоих протоколов запроса CMP и CRS, но не использует PKCS-10 в качестве формата сообщения запроса сертификата. Затем был разработан документ, определяющий формат сообщения управления сертификатом, для определения расширенного множества управляющих сообщений, которые передаются между компонентами PKI. Управляющие сообщения сертификата поверх CMS (CMC) были разработаны, чтобы позволить использовать существующий протокол (S/MIME) как протокол управления PKI, без необходимости разработки полностью нового протокола, такого как CMP. Он также включает PKCS-10 в качестве синтаксиса запроса сертификата, который используется после того, как проект документа CRS приостановлен.

Другие проблемы касаются отмены сертификата. Были разработаны многочисленные проекты документов, адресованные различным аспектам, относящимся к отмене сертификата. CMP поддерживает запрос отмены, ответ, аннонсирование отмены и запросы для сообщения CRL. CMC определяет

запрос отмены, ответ отмены и запросы для сообщений CRL, но использует CMS в качестве инкапсулирующего протокола. OCSP был разработан для решения проблем доверяющих групп, которые хотят обрабатывать проверку CRL для каждого CA в сертификационном пути. Он вводит on-line механизм для определения статуса указанного сертификата, что может предоставить более своевременную информацию об отмене, чем это возможно с CRLs. Был разработан простой протокол проверки сертификата (SCVP), который позволяет доверяющим группам передавать все, относящееся к проверке их сертификата, к другому участнику. WG аргументированно разделила те функции, которые должны поддерживаться, и те, которые должны быть ее собственным протоколом или включены в OCSP. В ответ был разработан проект документа, определяющий расширения OCSP, для включения функций SCVP. Для дальнейшей работы с расширениями OCSP были разработаны два проекта документа: делегированная проверка действительности пути (DPV) и делигированное обнаружение пути (DPD). В настоящий момент существуют по крайней мере три кандидата. Это OCSPv2, SCVP и DVCS.

3 Заключение

Важнейшим требованием к сетевым технологиям является требование интероперабельности продуктов и систем и возможность их бесшовной интеграции в распределенные инфраструктуры. Это решается стандартизацией форматов данных и протоколов взаимодействия и тестированием продуктов и систем на соответствие данным стандартам. В работе сделан анализ существующих стандартов в области VPN-технологий, что является основой для построения профилей VPN-технологий, которые в свою очередь обеспечивают высокую степень интероперабельности конечных продуктов.

Литература

[1] RFC 3280 «Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile».

[2] RFC 3281 «An Internet Attribute Certificate Profile for Authorization».

[3] RFC 2511 «Internet X.509 Certificate Request Message Format».

[4] RFC 2630 «Cryptographic Message Syntax».

[5] RFC 2797 «Certificate Management Messages over CMS».

[6] RFC 2510 «Internet X.509 Public Key Infrastructure Certificate Management Protocols».

[7] RFC 2559 «Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2».

[8] RFC 2560 «X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP».

[9] RFC 3379 «Delegated Path Validation and Delegated Path Discovery Protocol Requirements».

Применение нейронных сетей для решения задач кластеризации в процессе мониторинга информационной безопасности

В. В. Райх

В современных системах обнаружения вторжений (Intrusion Detection System - IDS) применяют две группы методов обнаружения компьютерных атак: основанные на знаниях и основанные на поведении [1].

Воплощением методов, основанных на знаниях, является сигнатурный анализ, базирующийся на тех же принципах, что и антивирусные средства: любой компьютерной атаке, как и разрушающему коду вируса, соответствует определенный сетевой трафик (для удаленной атаки) или поток системных событий (для локальной атаки) - сигнатура, фиксируемый средствами аудита и мониторинга и который отличает эту атаку от других атак или нормального, «законопослушного» поведения системы. Достоинствами этого способа выявления атак является высокая точность анализа, обусловленная тем, что ошибки первого рода полностью зависят от полноты набора сигнатур атак, а второго рода - от качества их составления, а также высокая скорость работы, являющаяся следствием простоты реализации операций сравнения данных мониторинга с содержанием набора (базы) сигнатур атак. К недостаткам же можно отнести возможность обнаружения только тех атак, сигнатуры которых имеются в базе IDS и, как следствие, трудоемкую и достаточно кропотливую процедуру составления сигнатур новых атак.

Методы, основанные на поведении, базируются на статистическом анализе. В данном случае поведение вычислительной системы характеризуется набором показателей, временные ряды значений которых собираются датчиками системы мониторинга. В дальнейшем значения показателей сравниваются с выявленными ранее значениями, соответствующими нормальной, штатной работе - профилем. Достоинства и недостатки в данном случае противоположны сигнатурному анализу. Скорость и точность обнаружения атак не так велика за счет большей вычислительной сложности и меньшей точности описания свойств работы контролируемой системы (особенно это касается ошибок второго рода). Сама процедура формирования и коррекции профилей по трудоемкости значительно превосходит процессы создания новых сигнатур. Вместе с тем, есть возможность зафиксировать ранее не встречавшиеся вторжения, т. к. любое отклонение от созданного профиля нормальной работы будет считаться нарушением защиты.

Одна из проблем, возникающих при реализации статистического анализа, заключается в разработке механизмов создания и коррекции профилей поведения контролируемой системы. В настоящее время перспективным считается направление, связанное с применением в этой сфере нейронных сетей, обладающих способностью самообучения. В контексте анализа статистических показателей с целью выявления компьютерных атак это означает способность самостоятельно формировать и адаптировать свое представление о нормальном поведении системы, чтобы в последующем фиксировать отклонения от него. При этом результатом анализа может быть как просто фиксация отклонения от штатного режима работы, так и распознавание вида (класса) обнаруженной компьютерной атаки. В настоящей работе будет рассмотрен способ формирования профилей поведения, представляющий собой решение задачи кластеризации многомерных векторов с использованием нейронных сетей для обнаружения аномального поведения в вычислительной системе. При этом аспект, связанный с подбором самих статистических показателей, значения которых требуется отслеживать, будет оставлен за скобками, так как его рассмотрение представляет отдельную научную задачу. Скажем только, что в зависимости от вида IDS, это могут быть показатели, относящиеся к сетевому трафику, системным ресурсам локального компьютера, последовательностям системных вызовов, генерируемых приложениями, функционирующими на локальном компьютере.

Совокупность значений статистических показателей, характеризующих состояние системы в каждый момент времени, образует вектор в некотором многомерном пространстве. В этом случае процесс функционирования системы может быть описан совокупностью точек в данном пространстве. Часть из них будут представлять нормальное, штатное поведение системы, другая часть - аномальное поведение, возникающее вследствие проведения компьютерной атаки.

В качестве гипотезы выдвинем предположение, что поведение любой защищаемой системы имеет свои закономерности, иначе применение статистических методов бессмысленно. Наличие таких закономерностей должно приводить к тому, что вектора в многомерном пространстве, описывающие состояния системы, будут образовывать скопления, соответствующие ее преимущественному поведению. Таким образом, все многомерное пространство представляется возможным разделить на области, часть из которых будет относится к нормальному поведению, а часть - к аномальному. Задача в такой постановке непосредственно относится к одной из самых распространенных областей применения нейронных сетей.

Ее решение возможно двумя способами: как распознавание образов и как кластеризация данных. Представляется, что применение кластеризации является более эффективным способом, если требуется различать только две категории: нормальное и аномальное поведение. В пользу этого утверждения можно привести следующие доводы.

1. Обучение нейросети требует наличия сбалансированной базы примеров. В случае распознавания образов необходимо представить как наборы векторов, относящихся к нормальному поведению, так и наборы противоположного свойства, для чего может потребоваться проведение атак на защищаемую систему с целью получения таких векторов. При этом базу примеров потребуется создавать для каждой защищаемой системы, что увеличивает трудоемкость внедрения IDS.

2. Большую часть времени любая система находится в штатном состоянии. Иными словами, большинство векторов, описывающих ее состояние, будет относится к «нормальным». Вследствие наличия закономерностей в работе системы указанные вектора должны образовывать кластеры в многомерном пространстве. Таким образом, применив некоторую процедуру фильтрации, можно простым наблюдением за контролируемой системой получить базу примеров для кластеризации и последующего получения профиля нормального поведения.

3. Кластеризация векторов, относящихся к штатному поведению защищаемой системы, с последующим анализом отклонения от сформированных профилей реализует более жесткую стратегию «что не разрешено, то запрещено».

Применение для решения задачи кластеризации именно нейронных сетей обусловливается тем, что не известны свойства многомерного пространства признаков и, в частности, количество кластеров в нем, тогда как большинство регулярных алгоритмов построены в том предположении, что такое число известно.

Из описанных в литературе нейронных сетей, которые применяются для кластеризации данных в условиях отсутствия сведений о свойствах многомерного пространства признаков, наиболее распространенными являются самоорганизующиеся карты Кохонена (Self-Organizing Map, SOM) и сети адаптивного резонанса, известные как нейропарадигмы ART1 (для бинарных векторов) и ART2 (для вещественных векторов) [2]. Каждая из них обладает своими достоинствами и недостатками.

Сеть Кохонена представляет собой двумерную решетку нейронов, веса которых в процессе обучения корректируются таким образом, чтобы каждый нейрон был в некоторой метрике (чаще в евклидовом пространстве) близок к максимальному числу обучающих примеров. При этом корректировке подвергается не только очередной нейрон, но и (хотя и в меньшей степени) окружающие его в двумерной решетке. По окончании обучения кластеры образуют векторы, отнесенные к одному нейрону. Иными словами, многомерное пространство покрывается гипешарами определенного радиуса, центрами которых являются нейроны сети. Все, что лежит за их пределами, таким образом, должно быть отнесено к аномальному поведению системы.

Недостатком сетей Кохонена является, прежде всего, невозможность дообучения в процессе работы, что очень важно для систем мониторинга, поскольку особенности поведения контролируемой системы могут со временем меняться. Как следствие, это приводит к необходимости сбора и хранения больших массивов данных для переобучения сети и временной задержке переобучения на время накопления новой обучающей выборки. Кроме того, не существует обоснованной процедуры определения радиусов гипершаров, покрывающих многомерное пространство. Наиболее распространенным способом здесь является использование расстояния до наиболее удаленного от центра вектора. С другой стороны, по причине невозможности дообучения данные сети устойчивы к «враждебному переобучению», когда часто повторяющиеся компьютерные атаки начинают восприниматься сетью как обычное состояние контролируемой системы. Кроме того, из-за отсутствия коррекции сети в процессе обработки очередного входного вектора процедура анализа выполняется с большей скоростью.

Сети адаптивного резонанса, в свою очередь, построены как постоянно дообучающиеся объекты. Они состоят из двух слоев нейронов, где в выходном слое веса нейронов также корректируются таким образом, чтобы к каждому нейрону относилось максимальное количество схожих (в некоторой метрике) векторов. При этом, если такого нейрона для очередного вектора не найдено, то он автоматически добавляется в сеть. Вследствие указанных особенностей сети адаптивного резонанса имеют, по сравнению с сетями Кохонена, противоположные достоинства и недостатки. С одной стороны они обеспечивают высокую адаптируемость к изменениям в поведении контролируемой системы, но с другой стороны подвержены враждебному обучению и являются менее быстродействующими (сети вещественных векторов ART2).

Таким образом, представляется, что построение нейросетевого алгоритма кластеризации, обеспечивающего и дообучаемость, и устойчивость к враждебному переобучению должно основываться на

комплексном использовании всех перечисленных выше нейропарадигм. В зависимости от исходных данных могут применяться как две сети (Кохонена и ART2), так и три (если существует возможность построения бинарных векторов для сети ART1). Рассмотрим оба варианта.

Алгоритм «SOM+ART2» (две сети):

1. Сбор векторов признаков для первичного обучения двух сетей.
2. Эксплуатация обученных сетей. Решение о соответствии входного вектора нормальному поведению принимается на базе результата работы сети ART2. Если этим решением вектор признан «нормальным», то он дополнительно сохраняется для последующего переобучения двух сетей. Если же он признан нормальным и сетью «Кохонена», то он разрешается для дообучения сети ART2. Таким образом, в процессе эксплуатации одна нейросеть (ART2) используется в роли основной, а вторая (Кохонена) - в роли ограничителя для исключения возможности враждебного переобучения.
3. При превышении заданного порога противоречий между результатами анализа сетями Кохонена и ART2 система в целом считается устаревшей и происходит ее переобучение (обеих сетей) на основе ранее накопленных данных.

Алгоритм «SOM+ART» (три сети):

1. Сбор векторов признаков для первичного обучения сетей Кохонена, ART1 и ART2.
2. Эксплуатация обученных сетей. Общее решение о соответствии вектора нормальному поведению принимается как согласованный результат работы сетей Кохонена и ART2. В случае противоречия между ними арбитром выступает сеть ART1, поскольку бинарные вектора (если их можно сформировать) являются производными от первичных статистических данных, использование которых наравне с вещественными векторами нецелесообразно. Если очередной вектор признается сетью Кохонена соответствующим нормальному поведению контролируемой системы, то данные разрешаются для дообучения сетей ART1 и ART2. Если общим решением вектор также признан нормальным, то он сохраняется для последующего переобучения всей системы.
3. При превышении заданного порога противоречий между результатами анализа сетями Кохонена и ART2 система в целом считается устаревшей и происходит ее переобучение (всех трех сетей) на основе ранее накопленных данных.

К настоящему времени проведены эксперименты по кластеризации данных, полученных на трафике локальной вычислительной сети, отдельно с использованием нейросетей Кохонена и ART2, показавшие жизнеспособность представленных в работе гипотез и положений. Предметом дальнейших исследований являются свойства описанных алгоритмов комплексного использования нейросетей для кластеризации данных, в частности, их скоростные характеристики и способы определения пороговых значений для фиксации момента переобучения.

Литература

[1] Лукацкий А. В. Обнаружение атак. 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2003. 608 с., ил.

[2] Базы данных. Интеллектуальная обработка информации. 2-е изд. / Корнеев В. В., Гареев А. Ф., Васютин С. В. и др. М.: Нолидж, 2001. 496 с.

Использование нейронных сетей для выявления и классификации атак в ОС UNIX

С. В. Васютин

Выявление аномального поведения привилегированных процессов является одним из способов обнаружения атак на вычислительную систему (ВС), функционирующую под управлением ОС UNIX [1]. Аномальное поведение может быть зафиксировано путем анализа последовательности системных вызовов (ПСВ), генерируемых привилегированными процессами. В работе [1] показано, что ПСВ может быть использована для обнаружения распространенных атак на ВС, функционирующую под управлением ОС UNIX.

В отличие от других распространенных ОС, некоторые версии UNIX включают в свой состав средства позволяющие фиксировать ПСВ, например, Basic Security Module для ОС Solaris. Версии со свободно распространяемым исходным кодом (ОС Linux) допускают модификацию ядра, позволяющую регистрировать ПСВ.

Одна из важнейших проблем, требующих решения в процессе создания системы обнаружения атак на основе анализа ПСВ, – поиск алгоритма, обеспечивающего обработку ПСВ в реальном времени.

Наиболее известен «словарный» подход, который заключается в следующем. Фиксируется последовательность системных вызовов $C = (c_1, c_2, \ldots, c_n)$, генерируемых привилегированным процессом и порожденными им процессами-потомками в условиях, отсутствия атак на ВС (здесь c_i – символ-идентификатор системного вызова [число системных вызовов в современных версиях ОС UNIX составляет около 300], n – количество вызовов, сгенерированных за время работы программы). С помощью «скользящего окна» последовательность C «нарезается» на слова длины l: $w_1 = (c_1, c_2, \ldots, c_l)$, $w_2 = (c_2, c_3, \ldots, c_{l+1}), \ldots, w_{n-l} = (c_{n-l-1}, c_{n-l-2}, \ldots, c_n)$. Длина l выбирается исходя из условий задачи и лежит обычно в пределах от 7 до 10 [2]. Полученные слова образуют словарь V.

ПСВ, генерируемая привилегированным процессом в процессе работы, формирует слова w^* длины l. Система обнаружения атак подсчитывает число слов w^*, отсутствующих в словаре V. Если это число удовлетворяет некоторому эмпирическому критерию, то гипотеза о наличии атаки отвергается [1].

Более интересным представляется подход, позволяющий не только обнаруживать факт аномального поведения привилегированного процесса, но и классифицировать тип атаки, вызывающий подобное поведение. В работе [2] для этого предлагается использовать несколько словарей, каждый из которых содержит слова, «типичные» для атак разных видов.

К основным недостаткам подходов, основанных на использовании словарей, следует отнести:

1) невозможность получения полного множества слов, генерируемых процессом, так как последовательность системных вызовов не является детерминированной;

2) значительный объем памяти, необходимый для хранения словарей.

Задача выявления и классификации атак на основе анализа ПСВ близка к задачам классификации и анализа временных рядов. Для их решения давно и успешно используются нейронные сети [3]. В работе [4] авторы предложили использовать рекуррентную нейронную сеть для выявления атак на основе ПСВ. Однако сложность обучения подобных сетей затрудняет их использование в тиражируемых системах обнаружения атак. Кроме того, подход к кодированию входных данных, предложенный в работе, не обеспечивает устойчивой работы системы при числе используемых системных вызовов большем 20–30.

Более стабильные результаты могут быть получены путем анализа частотных характеристик обращений процесса к системным вызовам. Исследование ПСВ, зафиксированных во время «нормальной» работы привилегированных процессов, а также ПСВ, сгенерированных атакованными процессами, позволяет утверждать, что распределение значений частот обращения к разным системным вызовам на отрезках ПСВ одинаковой длины для этих двух случаев различается. Это позволяет построить

классификатор, анализирующий частоту появления системных вызовов в словах ПСВ и принимающий решение о том, в каком режиме функционирует привилегированный процесс – нормальном или «аномальном».

Исходными данными для построения классификатора служат слова w_i длины l, полученные из ПСВ, где i изменяется в диапазоне от 1 до $(n-l)$, n – количество вызовов, сгенерированных за время работы программы. В каждом из слов w_i производится подсчет числа появлений системных вызовов и формируется вектор $Q_i = (q_1, q_2, \ldots, q_m)_i$, где q_j – количество вхождений вызова с идентификатором c_j в слово w_i, j изменяется в диапазоне от 1 до m, m – общее количество системных вызовов в системе. К каждому из векторов Q_i дополнительно добавляется идентификатор d_i, определяющий режим работы процесса, в котором получен вектор: «нормальный», «атака$_1$», «атака$_2$»,..., «атака$_k$» (где k – число атак известных видов, для которых обрабатывались ПСВ).

Множество пар $E = (Q_i, d_i)$ используется для построения процедуры классификации с помощью одного из известных методов, в том числе с применением факторного и дискриминантного анализа. В данной работе в качестве классификатора были использованы нейронные сети – группа непараметрических методов, позволяющих автоматически строить нелинейные модели на основе данных.

Для построения классификатора необходимо обучить нейронную сеть относить предъявляемые ей вектора Q_i к одному из режимов работы процесса, заданных идентификаторами d_i. Для того чтобы сеть научилась выявлять вектора, не относящиеся ни к одному из известных классов воздействий, используется следующий подход. Случайным образом генерируются вектора $Q_{\text{сл}}$, которые отстоят от каждого из имеющихся векторов Q_i на расстояние, не меньшее заданного в некоторой метрике. Эти вектора вместе с идентификатором $d_{\text{неизв}}$ (воздействие неизвестного типа) образуют пары, также используемые для построения процедуры классификации нейронной сетью.

Множество пар E, дополненное случайно сгенерированными парами, образует множество примеров для обучения нейронной сети. Таким образом, нейронная сеть обучается распознавать «нормальный» режим работы привилегированного процесса (когда ВС не подвергается атакам), ситуации, когда процесс подвергается воздействию атак известных видов, а также выявлять аномальное поведение процесса, вызванное неизвестными причинами.

В данной работе были использованы хорошо зарекомендовавшие себя при решении подобных задач нейронные сети с парадигмой многослойный персептрон, обучаемый методом обратного распространения ошибки. Этот вид сетей достаточно проработан теоретически – доказаны возможность аппроксимации любой измеримой функции с заданной точностью, сходимость алгоритмов обучения.

Обученная нейронная сеть способна в реальном времени классифицировать вектора Q^*, получаемые из слов w^*, генерируемых выполняющимся привилегированным процессом, выявляя «нормальный» режим работы, атаки известных видов, а также аномальное поведение процесса, вызванное неизвестными причинами. Скорость работы многослойного персептрона достаточно высока для того, чтобы выполнять классификацию в процессе работы ВС в реальном времени.

Экспериментальная проверка предлагаемого подхода проводилась на данных общедоступной базы, накопленной в University of New Mexico (URL – `http://www.cs.unm.edu/~immsec/data/`). Эта база содержит ПСВ, зафиксированные во время «нормальной» работы некоторых привилегированных процессов (sendmail, ps, login и др.), а также ПСВ, сгенерированные процессами под воздействием различных атак. База содержит данные, собранные с использованием ОС Linux и Solaris.

На основе «нормальных» ПСВ и ПСВ атакованных процессов формировались базы примеров для обучения нейронных сетей. Использовались сети с парадигмой многослойный персептрон. Для их обучения был применен модифицированный алгоритм обратного распространения ошибки, обеспечивающий добавление при необходимости нейронов к скрытому слою сети и прореживание межнейронных связей.

В экспериментах были использованы трехслойные сети с одним скрытым слоем. Число входных нейронов определялось исходя из общего количества системных вызовов, использованных привилегированным процессом в процессе нормальной работы и в «аномальных» режимах. Обычно это число находилось в диапазоне от 40 до 60. Количество выходных нейронов соответствовало числу возможных режимов работы процесса, включая «воздействие неизвестного типа» (от 3 до 6). Длина l слов w_i в процессе экспериментов находилась в диапазоне от 32 до 128. Выбор длины влиял на время обучения нейронной сети, а также точность классификации. С ростом длины слова время, затраченное на обучение нейронной сети, и точность возрастали. Время обучения нейронных сетей составляло от нескольких минут до нескольких часов в зависимости от длины слова, объема ПСВ и количества режимов работы привилегированного процесса. Обучение нейронных сетей производилось на компьютере

с процессором Intel Pentium III 800 МГц. Для реализации и обучения нейронных сетей использовалась библиотека НЕЙРОЭКСПЕРТ.

Результаты экспериментов показали высокую точность обнаружения известных атак (данные для которых были использованы при обучении нейронной сети). Обученная сеть смогла правильно классифицировать все ПСВ базы University of New Mexico.

Разработанная методика была реализована в программе SOSC (Sequence Of System Calls), предназначенной для обработки последовательности системных вызовов, генерируемой процессами и их потомками в Linux, с целью выявления аномального поведения наблюдаемых процессов, а также классификации видов воздействия, вызвавших аномальное поведение. SOSC функционирует совместно с STRACE – системой фиксации системных вызовов, входящей в стандартную поставку Linux.

Программа SOSC работает в трех режимах: накопления статистики, обучения и выявления аномалий. В режиме накопления статистики производится обработка ПСВ, генерируемых заданным процессом и его потомками, с целью накопления векторов Q_i, полученных в условиях отсутствия внешних воздействий, а также, если необходимо, при воздействиях известных типов. В режиме обучения множество пар $E = (Q_i, d_i)$ используются для обучения нейронной сети в соответствии с описанной выше методикой. В режиме выявления аномалий обученная нейронная сеть используется для классификации векторов-образов Q^*, полученных из ПСВ выполняющегося привилегированного процесса.

Анализ частоты использования системных вызовов не во временных интервалах, а в словах ПСВ, инвариантных относительно времени, позволяет накапливать статистику, обучать нейронные сети и использовать их для классификации воздействий на компьютерах с различными характеристиками производительности. Причем однажды обученная нейронная сеть легко тиражируется и дообучается при необходимости.

В процессе функционирования SOSC для фиксации ПСВ используется STRACE, однако метод отслеживания системных вызовов, используемый этой программой, неэффективен с позиции использования вычислительных ресурсов. Его применение ведет к существенному падению производительности вычислительной системы. В настоящее время проводятся работы по встраиванию механизма, реализующего нейросетевой подход к обнаружению и классификации атак, в ядро Linux, что позволит снизить вычислительные затраты при работе SOSC.

Литература

[1] Hofmeyr S. A., Forrest S., Somayaji A. Intrusion detection using sequences of system calls // Journal of Computer Security, 1998, Vol. 6, p. 151–180.

[2] Cabrera J. B. D., Lewis L., Mehra R. K. Detection and Classification of Intrusions and Faults using Sequences of System Calls // SIGMOD Magazine, Vol. 30., Number 4. December 2001.

[3] Корнеев В. В., Гареев А. Ф., Васютин С. В., Райх В. В. Базы данных. Интеллектуальная обработка информации. М.: Нолидж, 2001. 496 с.

[4] Ghosh A. K., Schwartzbard A., Schatz M. Learning Program Behavior Profiles for Intrusion Detection // Proceedings of the 1st Workshop on Intrusion Detection and Network Monitoring, April 9–12, 1999, Santa Clara, California, USA.

Анализ отдельных компонент трафика в системах активного аудита компьютерных сетей

В. А. Васенин, А. В. Галатенко, А. А. Макаров

1 Введение

Протоколирование и аудит является одним из основных сервисов программно-технического уровня обеспечения информационной безопасности компьютерных систем. Важная составляющая этого сервиса — активный аудит [1], основная цель которого заключается в выявлении (предпочтительно в режиме реального времени) злоумышленных или нетипичных действий с целью выработки оперативных адекватных мер противодействия. Использование активного аудита в общей системе управления распределенными информационно-вычислительными структурами способно существенно улучшить их защищенность.

Повышение эффективности систем активного аудита в значительной степени связано с использованием комплексного подхода. Такой подход объединяет результаты анализа данных сенсоров на разных уровнях структурной иерархии системы. Изначально, эти анализаторы могут быть построены по разному принципу и оперировать различными типами данных. Так, хостовые атаки трудно обнаружить чисто сетевыми методами. Распределенные по цели атаки могут быть выявлены только путем сопоставления результатов от нескольких хостовых анализаторов. Важной компонентой комплексной системы активного аудита должен стать анализатор верхнего уровня, следящий за «типичностью», «нормальностью» интегрального поведения большой распределенной системы.

В связи с изложенными обстоятельствам исследования и разработка подходов к совершенствованию механизмов мониторинга состояния подконтрольной системы и моделей анализа информации о ее состоянии на каждом из структурных уровней являются очень важными задачами.

Подчеркнем важность выявления именно нетипичных действий. За нетипичностью могут скрываться сбои в работе системы (вызванные как злоумышленниками, так и ошибками в программном обеспечении, администрировании или эксплуатации), реализация или последствия неизвестных атак, вход злоумышленника под украденным паролем и т. п. Используемые в большинстве как коммерческих, так и исследовательских систем активного аудита методы анализа на основе срабатывания правил не предназначены для выявления подобных аномальных ситуаций. Как следствие, эффективность таких систем далека от совершенства [1]. Учитывая стохастический характер большинства фиксируемых компонент трафика, для выявления его нетипичного поведения возникает необходимость привлечения методов математической статистики.

Большинство различных статистических методов анализа трафика (факторный, кластерный, дискриминантный анализ, нейронные сети) в своей основе, как правило, опираются и используют простейшие обобщенные характеристики различных компонент трафика, их средние значения, дисперсии, корреляции и т. п. Традиционные оценки этих величин без учета присущих трафику стохастических особенностей могут сильно искажаться. Последнее может заметно снижать эффективность выявление нетипичных ситуаций в сети. Учитывая это, в настоящей работе на основе анализа обширных данных о трафике в каналах связи за продолжительные сроки:

- сформулированы основные особенности стохастической изменчивости отдельных компонент трафика;
- предложена методика выделения участков локальной стационарности компонент трафика;
- предложена апробированная на реальных данных методика определения естественных границ вариации компонент трафика и выделения в нем нехарактерных значений.

2 Стохастические особенности компонент трафика

Процедуры мониторинга сетевого трафика извлекают данные из заголовков переданных (полученных) пакетов, агрегируя эти данные за определенное, фиксированное время. Обычно время агрегации может варьироваться от нескольких секунд до нескольких минут (все примеры в данной работе приводятся для времени агрегации пять минут). Вопрос выбора оптимального времени агрегации это отдельная содержательная задача, которая в данной публикации не рассматривается. В качестве основных первичных характеристик трафика, его компонент, обычно рассматривают: число байт, число пакетов, число соединений, совокупное время соединений, тип протокола на входе и выходе. Наряду с этими компонентами бывает полезно изучать и некоторые производные от первичных характеристик: средний размер пакета, среднюю скорость передачи данных и т. п. Стохастическая

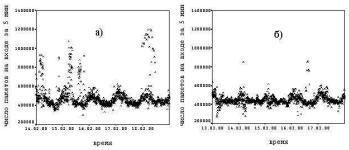


Рис. 1: Число пакетов за пять минут на входе канала в течение 5 рабочих дней: а) – данные 14-18.02.2000; б) – данные 13-17.03.2000.

природа большинства из этих компонент порождается самим механизмом передачи данных в компьютерных сетях и существенной гетерогенностью самой сетевой среды и пользовательской активности. В качестве основных источников этой гетерогенности выступают:

- техническая разнородность телекоммуникационной среды;
- разнородность сетевых приложений и протоколов передачи данных, обслуживающих эти приложения;
- различная интенсивность сетевой активности в разное время суток, будние, выходные и праздничные дни.

Изучение в течение продолжительных периодов времени трафика различных сетей [2, 3, 4, 5, 6] показывает, что сети по характеру их трафика можно разбить на несколько категорий. К первой из них относятся «формирующиеся» сети с малым, эпизодическим трафиком. Статистические методы анализа трафиков подобных сетей мало эффективны в силу хаотичности трафика и недостаточного объема содержательных наблюдений.

Ко второй категории можно отнести сети с «устойчиво возрастающим (убывающим)» трафиком. Модели типичного изменения трафиков подобных сетей описаны в [3]. Стохастические показатели трафика этих сетей содержат долговременные линейные или полиномиальные тренды и авторегрессионные компоненты. Учет этих особенностей трафика в задачах активного аудита позволяет увеличить однородность скорректированных наблюдений и повысить эффективность выявления нехарактерной сетевой активности.

Третья категория сетей включает сети с «относительно стабильным» трафиком. Подобная «относительная стабильность» может обеспечиваться однородным характером сетевых задач (в корпоративных сетях) или значительным числом разнородных сетевых задач и пользователей. «Относительно стабильный» трафик может порождаться и высокой загрузкой инфраструктуры сети и, в первую очередь, высокой загрузкой магистральных каналов. Анализ данных мониторинга подобных сетей показывает, что в компонентах их трафика можно выделить стационарные периоды. Наличие подобных периодов позволяет предложить устойчивые алгоритмы анализа компонент трафика и выявления в них нехарактерных значений.

Проиллюстрируем сформулированную идею на реальных данных мониторинга сети с «относительно стабильным» трафиком. В качестве компоненты трафика для примера рассмотрим число переданных пакетов на входе магистрального канала сети за пять минут.

На рис. 1а) и 1б) представлены графики числа переданных пакетов в течение 5-ти минут на входе канала Rbnet-Teleglobe (внешний канал в Интернет для Российской сети науки и образования) за две произвольно выбранные недели (без выходных дней) с месячным интер валом между неделями.

Из рис. 1а) и 1б) видно, что число передаваемых пакетов в единицу времени зависит от времени суток (даже при независимой от времени суток загрузке канала), и характер этой зависимости

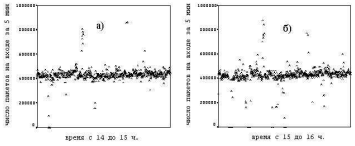


Рис. 2: Число пакетов за пять минут на входе канала на протяжение 3.5 месяцев: а) – данные с 14 до 15 час.; б) – данные с 15 до 16 час.

довольно устойчив. Кроме того, на фоне достаточно регулярного изменения во времени числа передаваемых пакетов наблюдаются отдельные нехарактерные наблюдения. Рассмотрим поведение этой характеристики трафика в течение фиксированного часа суток.

На рис. 2а) и 2б) приведено поведение числа переданных пакетов за пять минут в течение двух фиксированных часов в течение 3.5 месяцев, включая периоды наблюдений представленные на рис. 1.

Поведение числа переданных пакетов на рис. 2а) и 2б) представляется сходным как в течение фиксированного часа суток, так и в течение двух рассматриваемых часов. Однако для строго обоснования этого сходства необходимы формальные статистические процедуры, устойчивые к наблюдающимся на графиках 2а) и 2б) отдельным значительным отклонениям. (К примеру, сравнение средних значений и дисперсий подобных данных с помощью стандартных статистических критериев Стьюдента и Фишера часто приводят к неверным заключениям именно из-за наличия в данных нехарактерных значений.)

Сформулируем в общем виде основные характерные стохастические особенности компонент трафика.

1. Большинству компонент трафика присуща значительная (до 2–3 раз) внутрисуточная изменчивость даже при стационарной загрузке канала в течение суток. Основной причиной этой изменчивости выступает изменение протокольной структуры трафика в течение суток. Одним из главных показателей протокольной структуры может выступать доля протокола http в совокупном трафике. Для данных, приведенных на рис. 1 и рис. 2 эта величина равнялась приблизительно 70% в дневные часы и опускалась до 50% в ночные часы. Без учета возможной внутрисуточной изменчивости компонент трафика, значения являющиеся нехарактерными в одни часы суток могут ошибочно приниматься за характерные, так как они являются таковыми в другие часы суток.

2. Вероятностные распределения компонент трафика в фиксированный период времени суток, как правило, содержат небольшой процент наблюдений резко отклоняющиеся в обе стороны от области локализации основного массива наблюдений. Эти отклонения заметно искажают оценки дисперсий и корреляций компонент трафика поставляя заведомо ложную информацию в большинство многомерных методов анализа трафика. Они так же часто не позволяют установить согласие вероятностных распределений данных в различные периоды суток, так как поведение нехарактерных значений не согласованно.

3. Условные вероятностные распределения компонент трафика в фиксированное время суток, при условии удаления из данных нехарактерных значений (локализации данных в некоторых заданных границах) обычно не удается описать тем или иным известным параметрическим семейством распределений. Таким образом, отсутствует возможность определять характерность или нехарактерность значений наблюдений, исходя из оцененного по усеченным данным параметрического семейства распределений.

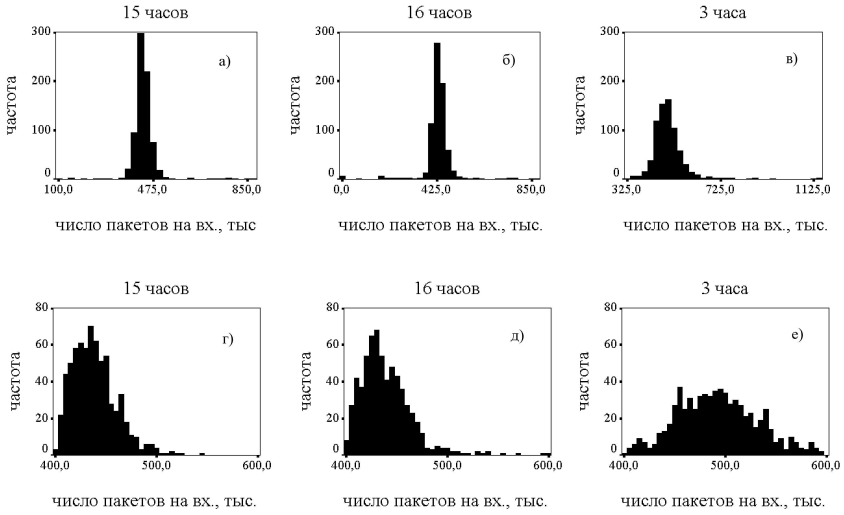


Рис. 3: Гистограммы исходных (а) – в)) и условных распределений (г) – е)) числа пакетов в различное время суток. Агрегированные за 5 мин. данные для входа канала Rbnet-Teleglobe за 3.5 месяца.

4. Условные вероятностные распределения компонент трафика даже после локализации их на естественных интервалах могут значительно отличаться в различное время суток. Другими словами компоненты трафика варьируют в течение суток не только свои среднее значение и дисперсию, но и характер распределения.

Проиллюстрируем перечисленные в пунктах 2–4 стохастические особенности компонент трафика на рис. 3.

На рис. 3а) и 3б) представлены гистограммы частот для данных приведенных на рис. 2а) и 2б) соответственно. Видно, что основная масса данных имеет сравнительно узкий интервал локализации, а небольшое число нехарактерных значений резко отклоняется от этого интервала в большую и меньшую стороны. На рис. 3г) и 3д) представлены гистограммы условных распределений для данных рис. 3а) и 3б), локализованные на более узком интервале наблюдений. Хорошо видна асимметрия этих распределений. Вид этих гистограмм позволяет надеяться на согласие условных распределений этих данных, соответствующих двум разным периодам суток. На рис. 3в) и 3е) представлены гистограммы исходных и условных распределений для данных, соответствующих 3 часам ночи. Сравнение этих распределений с распределениями на рис. 3а)–б) и 3г)–д) показывает возможные вариации распределений компонент трафика в течение суток.

Перечисленные выше стохастические особенности компонент трафика требуют накопления значительных объемов наблюдений (не менее нескольких сотен) для того, чтобы обоснованно судить о характерном или нехарактерном их поведении. Период наблюдений должен охватывать как минимум несколько недель или даже месяцев, а собранные данные мониторинга должны храниться в специальных базах данных. Сокращение периода агрегации данных, в принципе, увеличивает массивы наблюдений, но при этом может происходить потеря устойчивости распределений наблюдений, особенно в сетях с относительно небольшим трафиком. Однако ситуация значительно улучшается за счет того, что внутри суток можно выделить периоды времени различной длинны, на протяжении которых компоненты трафика или скорректированные компоненты с учетом трендов ведут себя стационарно. Именно на этом положении основана предлагаемая методика выделения нехарактерных значений компонент трафика.

3 Методика определения естественных границ вариации компонент трафика и выделения в нем нехарактерных значений

В основу методики выявления участков локальной стационарности компонент трафика с учетом их стохастических особенностей положен двухвыборочный критерий согласия распределений Колмогорова-Смирнова [7]. Этот критерий не предполагает априорных знаний о характере сравниваемых распределений и не требует в отличие от критериев согласия типа хи-квадрат разбиения данных на некоторые интервалы группировки. С учетом того, что исходные данные могут содержать нехарактерные значения, мы предлагаем применять этот критерий не только к исходным данным и их выборочным функциям распределения, но и к последовательно цензурируемым данным и порождаемым ими условным выборочным распределениям.

Обозначим через ξ и ξ' случайные величины, соответствующие выбранной компоненте трафика в различные стационарные периоды суток. Пусть $F(x)$ и $G(x)$ — их неизвестные функции распределения, а $F_n(x)$ и $G_m(x)$ — их эмпирические аналоги, построенные по выборкам. (Учитывая, что время соединения длится в среднем несколько секунд, а время агрегации данных составляет пять минут, можно считать наблюдения компонент трафика на соседних интервалах времени практически независимыми.) Рассмотрим условные функции распределения этих случайных величин $F(x \mid \xi \in [a_k, b_k])$ и $G(x \mid \xi' \in [a_k, b_k])$, при условии, что они попадают в один и тот же отрезок $[a_k, b_k]$. Обозначим через $F_{n_k}(x \mid \xi \in [a_k, b_k])$ и $G_{m_k}(x \mid \xi' \in [a_k, b_k])$ эмпирические функции соответствующих условных распределений. Пусть $\alpha_{n_k,m_k}(a_k, b_k)$ минимальный уровень значимости двухвыборочного критерия согласия Колмогорова-Смирнова D_{n_k,m_k}, примененного к усеченным выборкам против двустороней альтернативы неравенства условных распределений $F(x \mid \xi \in [a_k, b_k])$ и $G(x \mid \xi' \in [a_k, b_k])$. Ясно, что $\alpha_{n_k,m_k}(a_k, b_k)$ есть функция от границ усечений выборок a_k и b_k, которые существенно влияют на объемы усекаемых выборок и наличие в выборках наблюдений, значительно отклоняющихся от границ локализации основных массивов наблюдений. В том случае, если в исходных выборках содержится некоторое количество нехарактерных значений, при том, что распределения основной массы наблюдений совпадают, можно рассчитывать, что найдутся такие значения a_k и b_k, которые отсекут в каждой из выборок нехарактерные значения, а на остальных данных двухвыборочный критерий согласия Колмогорова-Смирнова обнаружит устойчивое согласие условных распределений $F(x \mid \xi \in [a_k, b_k])$ и $G(x|\xi' \in [a_k, b_k])$. То есть функция $\alpha_{n_k,m_k}(a_k, b_k)$ может быть использована для оценки неизвестных границ усечения. Пусть система отрезков $[a_k, b_k]$ является вложенной, то есть для любого $k : [a_{k+1}, b_{k+1}] \in [a_k, b_k]$. В сформулированной выше модели одинаково условно распределенных выборок на некотором неизвестном интервале при приближении границ интервала усечения к границам этого интервала минимальный уровень значимости $\alpha_{n_k,m_k}(a_k, b_k)$ должен достигать выбранного уровня значимости для принятия нулевой гипотезы и устойчиво превышать его при дальнейшем усечении. Следовательно, точки a_k и b_k, начиная с которых наблюдается подобное поведение функции $\alpha_{n_k,m_k}(a_k, b_k)$ являются оценками границ усечения данных. Они разбивают данные в двух сравниваемых выборках на две части. Часть, в которой поведение выборок согласовано и оставшиеся наблюдения, которые характеризуются, как не характерные для второй выборки с точки зрения первой (и обратно). Участки компонент трафика обладающие подобными свойствами будем называть локально стационарными. Очевидно, что из согласия условных распределений наблюдений на этих участках вытекает равенство их описательных статистик: среднего значения, медианы, дисперсии, стандартного отклонения и т. п. Значения, не попадающие в интервалы согласованного поведения компонент трафика, будем называть нехарактерными. Очевидно, что предложенная методика способна принести результаты только в том случае, если в трафике (или в скорректированном с учетом трендов трафике) действительно обнаруживаются периоды времени на которых основная масса наблюдений ведет себя согласованно. Практический анализ трафика магистральных каналов различных сетей подтверждает это допущение. При этом длительность во времени локально стационарных участков внутри суток колеблется от 3–4 часов до 20 минут. Примером локально стационарных данных могут служить данные о числе переданных пакетов на входе канала, представленные на рис. 1. В качестве первой выборки, в описанной выше методике, например, выступают данные фиксируемы с 14-00 до 15-00, а в качестве второй — данные с 15-00 до 16-00. Сходство гистограмм этих данных на отрезке [400–500] тыс. пакетов довольно хорошо заметно на рис. 2г) и 2д).

Для выбора вложенной последовательности отрезков $[a_k, b_k]$ можно использовать различные алгоритмы. Не вдаваясь в детали, укажем примерную схему одного из таких итеративных алгоритмов,

который может быть использован для автоматического проверки условного согласия двух выборок и вычисления границ интервала, отсекающего из выборок нехарактерные значения.

1. Объединить наблюдения двух рассматриваемых выборок в одну. Для полученной выборки вычислить базовые описательные статистики: минимум (min), максимум (max), медиану (med) и межквартильный размах (range).

2. Задать число шагов отсечения n и максимально допустимый процент удаляемых данных из каждой выборки (part). Вычислить размер шага усечения данных сверху (stepup) и снизу (steplow):

$$\text{stepup} = (\max - (\text{med} + 1.5 * \text{range}))/n,$$
$$\text{steplow} = ((\text{med} - 1.5 * \text{range}) - \min)/n.$$

3. Определить границы отрезка $[a_k, b_k]$ для очередного шага k:

$$a_k = \min + (k-1) * \text{steplow},$$
$$b_k = \max - (k-1) * \text{stepup}.$$

4. Для каждой выборки вычислить процент наблюдений, не попавших в отрезок $[a_k, b_k]$. Если хотя бы в одной из выборок эта величина превышает допустимый процент удаляемых данных, то итеративная процедура прерывается. Если при этом на предшествующем шаге не было достигнуто согласие условных распределений двух выборок при заданном уровне значимости принятия гипотезы, то распределения выборок считаются условно не согласованными.

5. На отрезке $[a_k, b_k]$ вычислить минимальный уровень значимости $\alpha_{n_k,m_k}(a_k, b_k)$ статистики критерия D_{n_k,m_k} и сравнить его с выбранным уровнем значимости α. Если $\alpha_{n_k,m_k}(a_k, b_k) > bk) > \alpha$ то величины a_k, b_k могут рассматриваться как первичное приближение для оценок границ цензурирования. Они могут уточняться за счет уменьшения шагов усечения данных, начиная с $(k-1)$ шага.

Изложенная примерная схема может привести либо к нахождению границ отрезка $[a_k, b_k]$, на котором условные распределения выборок согласованы, либо к фиксации отсутствия согласия условных распределений (при заданных требованиях к максимально допустимому проценту цензурирования и уровню значимости критерия). В последнем случае, если речь идет об анализе компоненты трафика, следует изменить условия формирования одной или двух выборок для анализа, например, рассмотреть данные за более короткие отрезки времени. На рис. 4. приведен протокол работы описанной процедуры для сравнительного анализа условных распределений числа передаваемых пакетов за пять минут для двух различных часов суток.

Из протокола видно, что превышение 5-ти процентного уровня значимости статистикой Колмогорова-Смирнова происходит на 12-ом шаге. При этом границы отсечения равны $a_k = 2096$, $b_k = 41768$, а совокупный процент отсечения составил 1.1%.

4 Методика выделения участков локальной стационарности компонент трафика внутри суток

Кратко опишем один из возможных схем выделения участков стационарности компоненты трафика внутри суток.

1. Разбить данные трафика внутри суток на часовые отрезки.
2. Объединить данные соответствующие фиксированному часу в одну выборку за весь период наблюдений.

час	час	границы		статистика	уровень	процент отсечения		
		мин	мах		значимости	в целом	снизу	сверху
2	6	1004	49797	1,421724	0,035	0	0	100
2	6	1103	49067	1,480169	0,025	0,6	0,2	99,6
2	6	1203	48337	1,459563	0,028	0,6	0,2	99,6
2	6	1302	47607	1,459563	0,028	0,6	0,2	99,6
2	6	1401	46878	1,459563	0,028	0,6	0,2	99,6
2	6	1501	46148	1,438906	0,032	0,6	0,2	99,6
2	6	1600	45418	1,438906	0,032	0,6	0,2	99,6
2	6	1699	44688	1,438399	0,032	0,7	0,2	99,5
2	6	1798	43958	1,396875	0,04	0,7	0,2	99,5
2	6	1898	43228	1,374317	0,046	0,7	0,2	99,5
2	6	1997	42498	1,374851	0,046	1	0,2	99,3
2	6	2096	41768	1,311171	0,064	1,1	0,2	99,1
2	6	2196	41039	1,290036	0,072	1,1	0,2	99,1
2	6	2295	40309	1,246839	0,089	1,2	0,2	99
2	6	2394	39579	1,225466	0,099	1,5	0,2	98,8
2	6	2494	38849	1,203031	0,111	1,8	0,2	98,4
2	6	2593	38119	1,136525	0,151	1,8	0,2	98,4

Рис. 4: Протокол процедуры проверки согласия условных распределений двух выборок

3. При отсутствии в полученных почасовых выборках трендов и циклических компонент, для каждого из 276-ти сочетаний пар часов провести сравнительный анализ условных распределений соответствующих им выборок с помощью алгоритма, описанного выше. Выделить все пары условно согласованных данных и часы, данные которых, не согласуются ни с одним другим часов.

4. При обнаружении нескольких выборок внутри суток с попарно согласованными условными распределениями найти общие согласованные границы локализации для всех этих часов.

5. Для часов, не имеющих согласованных пар, разбить их на 3 или 4 части и повторить поиск согласованных интервалов времени для вновь выделенных периодов. Приведенная выше примерная схема может быть оптимизирована, и перебор всех возможных пар часов существенно сокращен. Для этого следует на предварительном этапе с помощью устойчивых оценок центров и разбросов выборок выделить заведомо не согласованные выборки и исключить их из сравнительного анализа. Однако в настоящей работе эти и другие возможности оптимизации не рассматриваются.

Результаты расчетов для различных сетей показывают, что в компонентах их трафика внутри суток можно выделить несколько (от 3-ех до 5-ти) продолжительных периодов условной стационарности внутри суток. Длительность таких периодов колеблется от 2-х до 5-ти часов. Например, для данных числа переданных пакетов на рис. 1 было выделено 5 стационарных долговременных периодов, самый длинный из которых включал 4 часа и продолжался с 14-00 до 18-00. Часть данных этого периода приведена на рис. 2.

5 Выводы

Анализ трафика компьютерных сетей не простая, но необходимая составляющая систем активного аудита. Он направлен на выявление в трафике нехарактерных ситуаций и способствует повышению защищенности сети.

При использовании различных статистических методов и алгоритмов для выявления нехарактерных ситуаций в трафике необходимо учитывать особенности стохастической изменчивости трафика внутри суток и предварительно очень внимательно исследовать используемый метод на адекватность и устойчивость к особенностям трафика.

Для существенного повышения точности и достоверности выводов следует добиваться сравнения однородных (условно одинаково распределенных) данных трафика.

Предложенная в статье оригинальная методика, не претендуя на полную универсальность (в изложенном виде), позволяет выделять в компонентах трафика локально однородные периоды различной продолжительности и одновременно определять границы локализации характерной части трафика.

Апробация методики на данных различных сетей показало ее продуктивность.

Литература

[1] Галатенко А. В. Активный аудит // Jet Info, М.: «Джет Инфо Паблишер» № 8(75), 1999, с. 1–28

[2] Васенин В. А. Российские академические сети и Internet // М.: РЭФИА, 1997 - 174 с.

[3] Макаров А. А., Симонова Г. И., Ковба Н. Л. Закономерности изменения загрузки магистральных каналов компьютерных сетей // Автоматика и телемеханика, № 12, 2000, с. 104-114

[4] Макаров А. А., Ковба Н. Л., Турков В. А. Структура трафиков научно-образовательных сетей России на канале RBnet-Teleglobe // Материалы Междунар. науч.-методич. конф. «Новые информационные технологии в университетском образовании». Новосибирск, 2000., с. 136.

[5] Макаров А. А., Симонова Г. И. Статистическая модель внутрисуточных колебаний скорости передачи данных пользователям компьютерных сетей // Статистические методы оценивания и проверки гипотез. Межвуз. Сб. научных трудов, Пермь, вып. 14, 2001 с. 158–169.

[6] Макаров А. А., Симонова Г. И., Ковба Н. Л., Турков В. А. Стохастические модели мониторинга телекоммуникационных сетей // Вестник Херсонского государственного технического университета, № 3(12). Херсон, 2001, с. 168–171.

[7] Большев Л. Н., Смирнов Н. В. Таблицы математической статистики // М.: Наука, Гл. ред. физ.-мат. лит., 1983, 416 с.

Статистическая модель обнаружения одного класса удаленных сетевых атак в высокоскоростных компьютерных сетях

Н. О. Вильчевский, М. Б. Гайдар, В. С. Заборовский, В. Е. Клавдиев

Аннотация

Рассматриваются вопросы организации защиты от удаленных атак на элементы компьютерной сети в момент их исполнения. Предлагается аналитическая модель расчета вероятности успешного отражения атаки для различных видов функции распределений.

Ключевые слова: Ключевые слова: атака, TCP-соединение, распределение вероятностей, пакет, хакер.

1 Введение

Работа посвящена проблеме обнаружения удаленной атаки типа «подмена доверенного субъекта TCP соединения». Выбор типа атаки определяется следующими обстоятельствами. Успех атаки выбранного типа наиболее вероятен, если поведение интервента в некотором (из далее приведенных рассуждений будет ясно, в каком смысле употреблен этот термин) смысле оптимально. А это, в свою очередь, требует от интервента совершенно определенной манеры поведения, что делает его «заметным». В свою очередь, это дает неплохие шансы обнаружить атаку еще на этапе ее исполнения, а не постфактум, как это, к сожалению, бывает чаще всего.

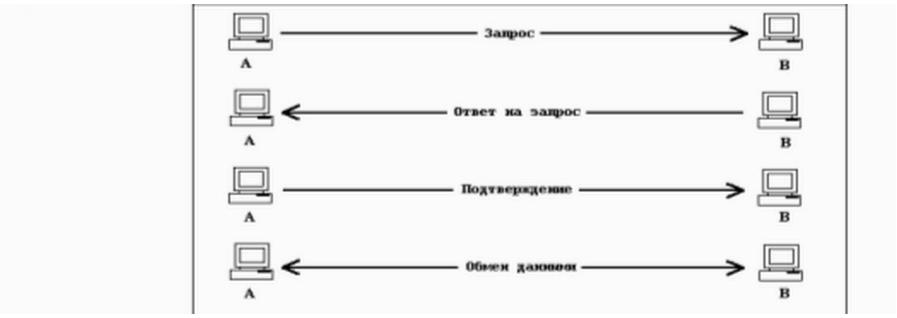


Рис. 1: «Рукопожатие».

Рассмотрим способ организации атаки [1]. Интервент осуществляет подмену одного из доверенных субъектов диалога в момент «рукопожатия», которое выглядит так, как это показано на рисунке 1.

Атакующий, планируя выдать себя за субъект A, должен обезопасить себя от вмешательства A в диалог и отвечать B именно «теми словами», которые должен был бы произносить A при «рукопожатии». Т. е., атака должна выглядеть так, как это показано на следующем рисунке 2.

Атаке должна предшествовать очень важная акция — «разведка», во время которой интервент выясняет, какие именно «слова» он должен произнести, выдавая себя за субъект A, т. е., выясняется закон формирования имен пакетов, присущих именно этому организуемому диалогу. Схема «разведки» проста — это несколько законных запросов на соединение от своего собственного имени. На этом этапе хакер, получая чрезвычайно важную для себя информацию, никаким образом себя в сети не обнаруживает.

Перейдем к более формальному рассмотрению поставленной задачи [2, 3].

Определение 1. Будем называть разведкой (подготовкой к атаке) посылку хостом X на хост B последовательности пакетов с целью определения вероятностной характеристики значения идентификационного номера, присваиваемого хостом B сообщению, в момент T — времени планируемой атаки.

Определение 2. Будем называть началом атаки посылку в момент времени T хостом X на хост B от имени хоста A запроса на открытие соединения.

Определение 3. Будем называть атакой отправку хостом X на хост B двух последовательных, размером K, серий пакетов в моменты времени T_1 и $T_1 + u$ соответственно $(T_1 > T, u \geqslant 0)$. Причем, номера пакетов в серии соответствуют предполагаемому номеру пакета, начавшего атаку. Будем далее обозначать через $\overline{k}$ множество номеров пакетов в сериях.

Определение 4. Будем называть атаку успешной при выполнении условий:

a) в каждой серии пакетов, посылаемых во время атаки, содержится пакет с номером, связанным с номером пакета, начавшим атаку;

b) пакет, с номером, связанным с номером пакета, начавшим атаку, во второй серии пришел на хост B после аналогичного пакета в первой серии, но раньше некоторого предельного времени $T_1 + \overline{u}$.

Пусть $F(t)$ —- функция распределения длительности задержки пакета в сети, т. е., вероятность того, что время доставки пакета от отправителя до получателя не больше t. Очевидно, что $F(t) = 0$ для $t \leqslant 0$.

Пусть P_k — вероятность того, что пакет с нужным идентификационным номером содержится в посылаемых сериях пакетов, т. е., во множестве $\overline{k}$.

Посмотрим, чем обеспечено это условие. Рассмотрим более детально *этап разведки*. Будем считать, что присвоение идентификационных номеров пакетам осуществляется хостом B последовательно в соответствии с некоторой линейной функцией времени. Для аппроксимации этой функции используем метод двух пробных («тестовых») сигналов, посылаемых в моменты времени t_1 и t_2. Пусть сама атака осуществляется (начинается) в момент времени $T_{\text{ат}}$. Без ограничения общности, примем

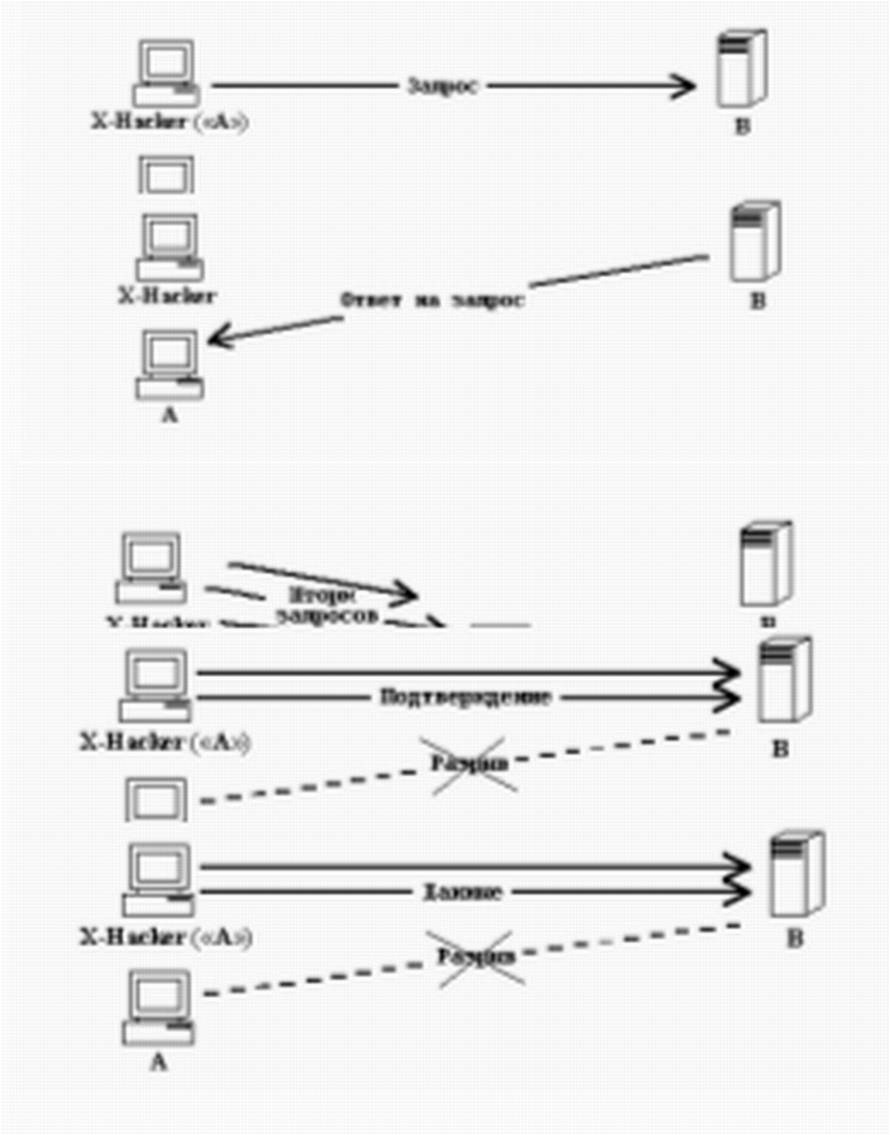

Запрос
B
X-Hacker («A»)
X-Hacker
Ответ на запрос
B
A
Поток запросов
Подтверждение
X-Hacker («A»)
B
Данные
X-Hacker («A»)
B
A

Рис. 2: «Атака».

за начало отсчета времени $t_1 = 0$. Обозначим за t — момент посылки второго тестового пакета и за T — начало атаки.

В результате посылки пробных сигналов, хост X получает информацию об идентификационных номерах, присвоенных этим сигналам. Обозначим их за N_1 и N_2 соответственно. Обозначим за τ_1 и τ_2 — запаздывания этих сигналов в сети при их передаче хосту B. Тогда, в силу предположения о линейной зависимости присвоенного пакету идентификационного номера от момента его прихода, получаем, что запросу, посланному в момент начала атаки T, будет присвоен номер, определяемый формулой

$$N^* = N_1 + K * \frac{T - \tau_1}{t + \tau_2 - \tau_1}. \tag{1}$$

В то же время, расчетное (прогнозируемое) хостом X значение этого номера:

$$N = N_1 + K * \frac{T}{t}. \tag{2}$$

Здесь $K = N_2 - N_1$.

Рассмотрим условия, при которых «разведка» может дать достоверные результаты, а именно:

a) начало атаки состоится после получения ответов на оба тестовых запроса —

$$T - \tau_1 > 0, \quad T - t - \tau_1 > 0; \tag{3}$$

b) $N_2 > N_1$ — это требование равносильно тому, что второй пробный пакет пришел на хост B позже первого, т. е.:

$$t + \tau_2 - \tau_1 > 0. \tag{4}$$

Обозначим область значений (τ_1, τ_2), удовлетворяющую условию (3)–(4) и очевидному условию $\tau_1 \geqslant 0$, $\tau_2 \geqslant 0$ через S_0. Нетрудно видеть, что

$$S_0 = \begin{cases} 0 \leqslant \tau_2 \leqslant T - t, \\ 0 \leqslant \tau_1 \leqslant t + \tau_2. \end{cases} \tag{5}$$

Качество результата «разведки» будем характеризовать близостью фактического значения идентификационного номера к прогнозируемому. Для этого рассмотрим неравенство

$$N^* - N \leqslant \Delta, \tag{6}$$

при выполнении условий принадлежности (τ_1, τ_2), множеству S_0. В силу (1)–(2) эта система неравенств имеет вид:

$$\begin{cases} K * \left(\dfrac{T - \tau_1}{t + \tau_2 - \tau_1} - \dfrac{T}{t} \right) < \Delta, \\ (\tau_1, \tau_2 \in S_0). \end{cases} \tag{7}$$

Решение этой системы неравенств дается формулами:

$$\begin{aligned} \Delta \geqslant 0 \implies S_1 &= \begin{cases} 0 \leqslant \tau_2 \leqslant T - t, \\ 0 \leqslant \tau_1 \leqslant \dfrac{(TK + \Delta t)\tau_2 + \Delta t^2}{TK + \Delta t - tK} \end{cases}; \\ -K(T - t) < -\Delta t < 0 \implies S_2 &= \begin{cases} -\dfrac{\Delta t^2}{TK + \Delta t} \leqslant \tau_2 \leqslant T - t \\ 0 \leqslant \tau_1 \leqslant \dfrac{(TK + \Delta t)\tau_2 + \Delta t^2}{TK + \Delta t - tK} \end{cases}; \\ \Delta < -K(T - t) \implies &\text{ решений нет.} \end{aligned} \tag{8}$$

Основываясь на полученных выше результатах, можно сформулировать следующую теорему:

Теорема 1. *Пусть запаздывания в сети являются независимыми случайными величинами с функцией распределения $F(\tau)$. Тогда функция распределения случайной величины $N^* - N$, при условии, что $(\tau_1, \tau_2 \in S_0)$, определяется зависимостью:*

$$P(\Delta) = \begin{cases} \dfrac{\int_0^{T-t}\left[\int_0^{\frac{(TK+\Delta t)\tau_2+\Delta t^2}{TK+\Delta t-tK}} dF(\tau_1)\right] dF(\tau_2)}{\int_0^{T-t}\left[\int_0^{t+\tau_2} dF(\tau_1)\right] dF(\tau_2)} & \text{при } \Delta > 0; \\ \int_{\frac{\Delta t^2}{TK+\Delta t}}^{T-t}\left[\int_0^{\frac{(TK+\Delta t)\tau_2+\Delta t^2}{TK+\Delta t-tK}} dF(\tau_1)\right] dF(\tau_2) & \text{при } -\frac{K}{t}(T-t) < \Delta < 0; \\ 0 & \text{при } \Delta < -\frac{K}{t}(T-t). \end{cases} \tag{9}$$

Замечание. Как в (9), так и в неравенствах (8), не учитывается целочисленность присваиваемых идентификационных номеров. Это, однако, вполне допустимо, учитывая то, что, как правило, эти номера имеют достаточно большие величины.

Оценив вероятность того, что серия атакующих пакетов содержит пакет с нужным идентификационным номером, сформулируем следующую теорему:

Теорема 2. *Вероятность успеха атаки определяется формулой:*

$$P = P_k * \left(F(\overline{u}) * F(\overline{u} - u) - \int_u^{\overline{u}} F(x-u)\, dF(x) \right). \tag{10}$$

Доказательство. Действительно, пусть $P_k(s)$ — вероятность того, нужный идентификационный номер равен s. Тогда вероятность того, что этот пакет удовлетворит условию b) определения 4:

$$P = \int_{T_1}^{T_1+\overline{u}} \left(\int_x^{T_1+\overline{u}} dF(y - T_1 - u) \right) dF(x - T_1),$$

где x — момент прихода пакета с s-м идентификационным номером, а y — время прихода пакета с нужным номером во второй серии. Положив $x - T_1 = \tau_1$, $y - T_1 = \tau_2$ и вычислив внутренний интеграл, имеем

$$\begin{aligned} P &= \int_0^{\overline{u}} \left(F(\overline{u}) - F(\tau_1 - u)\right)\, dF(\tau_1) \\ &= F(\overline{u}) * \left(F(\overline{u} - u) - F(0)\right) - \int_0^{\overline{u}} F(\tau_1 - \tau)\, dF(\tau), \end{aligned}$$

а учитывая, что $F(t) = 0$ при $t \leqslant 0$, получим

$$P = \left(F(\overline{u}) * F(\overline{u} - u) - \int_u^{\overline{u}} F(\tau_1 - u)\, dF(\tau_1) \right).$$

Таким образом, вероятность успешной атаки, в предположении, что нужный идентификационный номер имеет s-ый пакет, равна

$$P_k(s) = p_k(s) * P.$$

Суммируя по всем $s : P_k = \sum s \in kP_k(s)$, получим требуемую формулу (10). □

Для практического применения результатов проведенных исследований приходится конкретизировать вид функции распределения вероятности $F(t)$. Она может оказаться в классе фрактальных, экспоненциальных и других законов. В приведенных далее и снабженных необходимыми комментариями, результатах некоторых проводившихся численных экспериментов, использован экспоненциальный закон распределения $F(t)$.

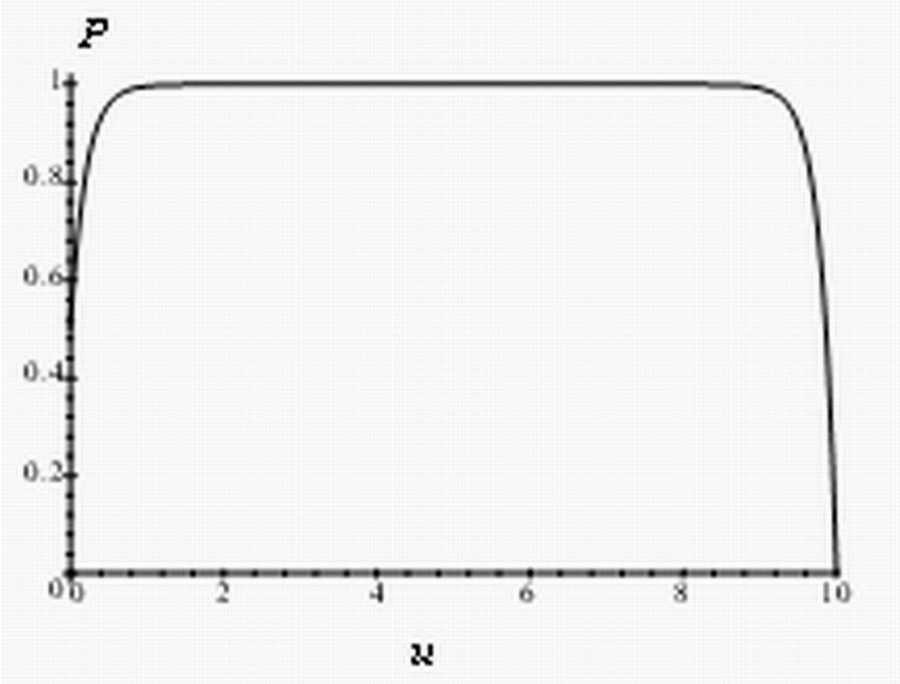


Рис. 3: Зависимость вероятности успеха атаки от времени между атакующими залпами.

2 Результаты численных экспериментов

Примем предположение, что

$$F(\tau) = 1 - e^{-\lambda\tau}, \quad \tau \geqslant 0, \quad \lambda = \frac{1}{t_{\text{ср}}},$$

где $t_{\text{ср}}$ — среднее время задержки в сети, тогда

$$P = P_k \left(1 - e^{-\lambda(\Delta - u)} + \frac{1}{2} e^{-\lambda(2\Delta - u)} - \frac{1}{2} e^{-\lambda u}\right). \tag{11}$$

и при $\lambda = 5$, $\Delta = 10$ имеем зависимость, показанную на рис. 3.

Если ввести минимальное время запаздывания пакетов в сети: u^*, получим экспоненциальный закон распределения со сдвигом, (11) примет вид:

$$P = P_k \left(1 - e^{-\lambda(\Delta - u - u^*)} + \frac{1}{2} e^{-\lambda(2\Delta - 2u^* - u)} - \frac{1}{2} e^{-\lambda u}\right),$$

а зависимость P от времени u между первой и второй атакующими сериями пакетов, которым может манипулировать хакер, при $\lambda = 5$, $\Delta = 10$, $u^* = 0.05$ представлена на рис. 4. Как видно, характер зависимости сохранился и позволяет сделать выбор u, обеспечивающий успех атаки при данных статистических параметрах.

Оценка вероятности того, что определенный в результате проведенной разведки идентификационный номер «нужного» пакета в атакующей серии будет отстоять от истинного, не более чем на Δ, получен из соображений

$$P(\Delta_{\max}) - P(\Delta_{\min}) \to \max,$$
$$\text{или } P(\Delta_{\min} - \Delta_N) - P(\Delta_{\min}) \to \max,$$

где $\Delta_{\max}$ и $\Delta_{\min}$ — максимальный и минимальный идентификационные номера пакетов в серии, а $\Delta_N = \Delta_{\max} - \Delta_{\min}$.

Здесь t — время между тестовыми пакетами при проведении разведки, T — момент посылки первой атакующей серии. Изменения параметров атаки, приводят к эффектам, приведенным на рис. 6, 7.

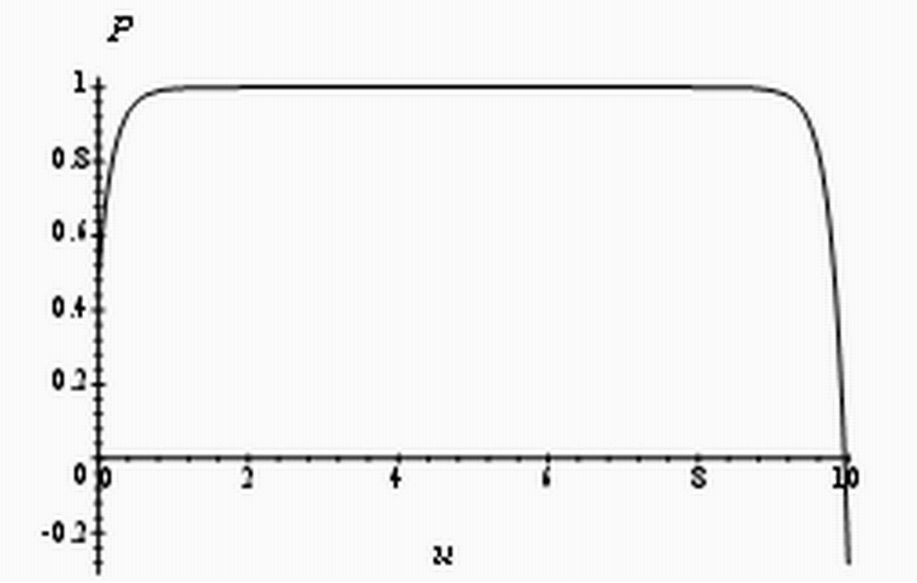


Рис. 4: Зависимость вероятности успеха атаки от интервала между атакующими залпами с учетом u^*.

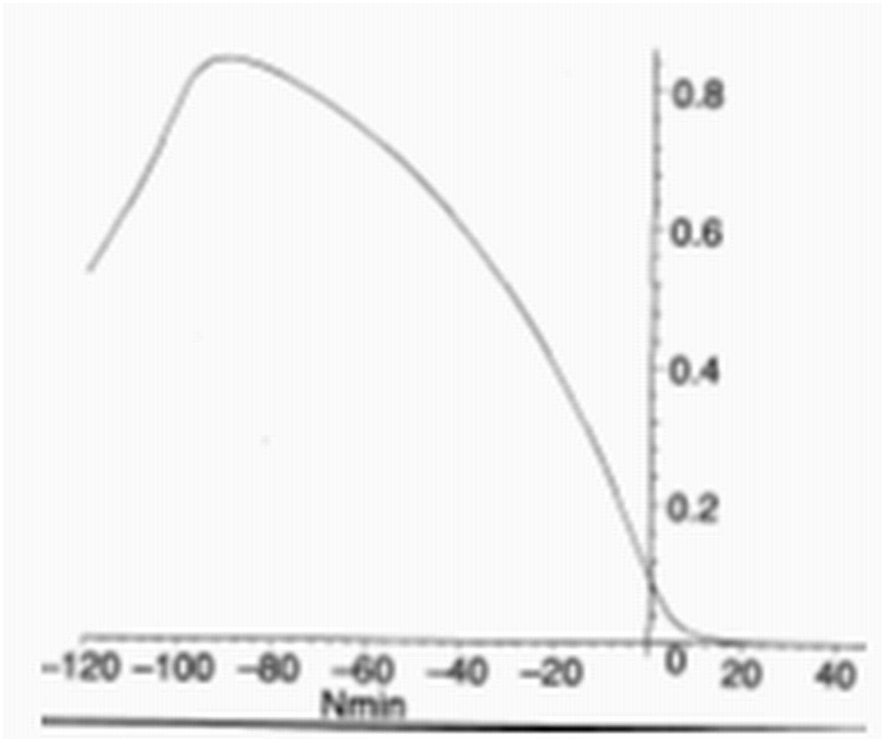


Рис. 5: Зависимость $P(\Delta_{\min} + \Delta_N) - P(\Delta_{\min})$ от N, $P = 0.843601$, $T = 10$, $t = 9$, $t_{\text{ср}} = 10$, $\Delta_N = 100$, $N_1 = 1000$, $N_2 = 5000$, $N_{\min} = -90$.

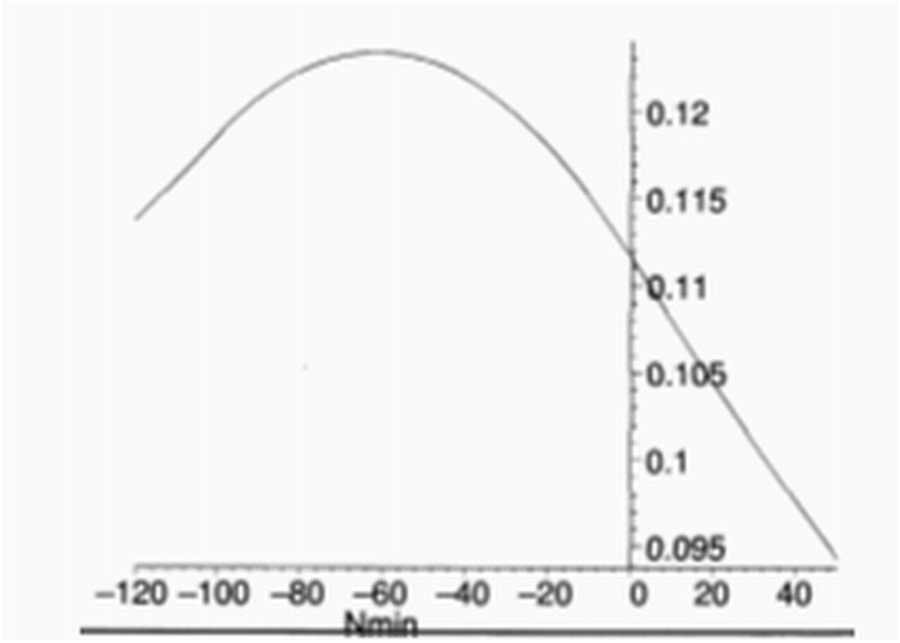


Рис. 6: $P = 0.123406$, $T = 10$, $t = 3$, $t_{\text{ср}} = 10$, $\Delta_N = 100$, $N_1 = 1000$, $N_2 = 5000$, $N_{\min} = -61$.

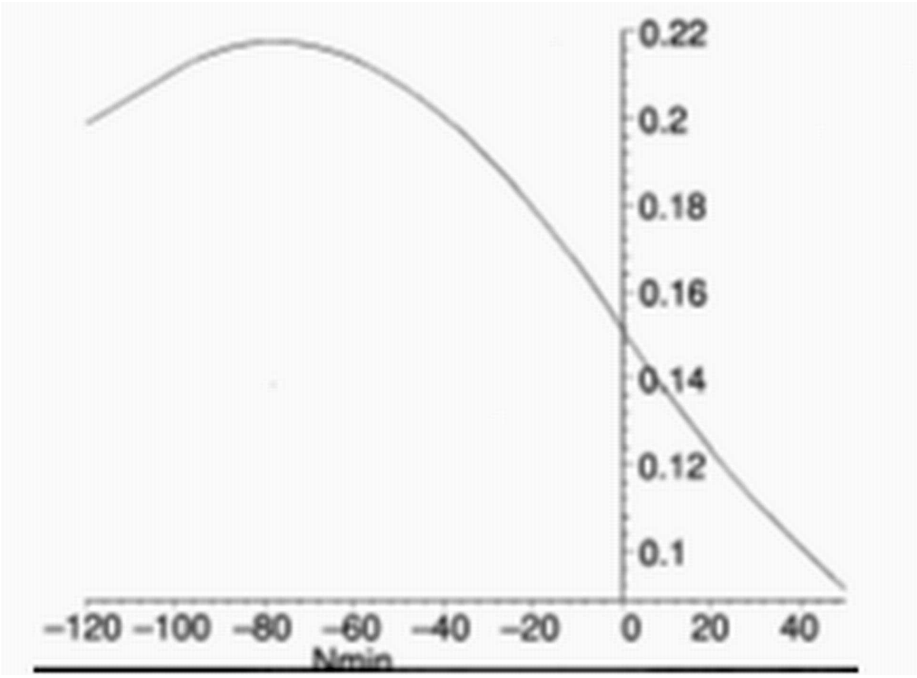


Рис. 7: $P = 0.217668$, $T = 3$, $t = 2$, $t_{\text{ср}} = 10$, $\Delta_N = 100$, $N_1 = 1000$, $N_2 = 5000$, $N_{\min} = -78$.

3 Заключение

Создание математических моделей сетевых атак позволяет повысить эффективность систем защиты. Точность идентификации атак зависит от возможности оперативной оценки параметров состояния сети (задержки, пропускная способность и пр.). Полученные результаты могут быть уточнены на основе экспериментальных оценок функций распределения, используемых в условиях реального функционирования компьютерных сетей.

Литература

[1] Медведовский И., Семьянов П., Платонов В. Атака через Internet. Санкт-Петербург, НПО «Мир и семья-95», 1997.

[2] Способ обнаружения удаленных атак в компьютерной сети. Патент № 2179738, приоритет: 24.04.2000. Авторы: Н. О. Вильчевский, В. С. Заборовский, В. Е. Клавдиев, В. А. Лопота, А. В. Маленкова.

[3] Klavdiev V. E., Vilchevsky N. O., Malenkova A. V. Development of tools for discovering and defence from attacks on remote hosts of the Internet. Int. Workshop NDTCS, 1999, St. Petersburg.

Анализаторы программного кода для комплекса «Тест»

В. Л. Олехов

Разрабатываемый в НИВЦ МГУ комплекс «Тест» предназначен для анализа исходных текстов и документации больших информационных систем, сопоставления заявленной и фактической функциональности исследуемого программного обеспечения.

Для обработки исходных текстов в комплексе «Тест» применяются анализаторы, которые разделяются на внутренние и внешние. Внутренние анализаторы являются неотъемлемой частью комплекса, в то время как внешние являются результатом разработок независимых организаций.

В комплексе «Тест» реализован внутренний анализатор для языка C. Работа анализатора заключается в исследовании исходных текстов и сохранении результатов во внутреннем представлении (ВП). ВП представляет собой систему связанных таблиц:

- таблица исходных файлов,
- таблица пространств имён,
- таблица типов,
- таблица переменных,
- таблица функций,
- таблица использований объектов.

В процессе разработки структуры ВП главной целью ставилось повысить скорость доступа к часто используемым элементам. Схема работы анализатора для языка C в целом совпадает с фазами работы обычного компилятора: предварительный лексический анализ, препроцессор, дополнительный лексический анализ и синтаксический анализ. В [1] фазами называются: лексический анализ, синтаксический анализ и генерация кода. Здесь фаза генерации кода отсутствует, однако ее элементы присутствуют в фазе синтаксического анализа (анализатор работает по так называемой синтаксически управляемой схеме: фаза выработки конечного результата отсутствует, основной вывод данных происходит во время синтаксического анализа).

Во время предварительного лексического анализа исходный текст разделяется на лексемы с указанием позиции каждой лексемы. Комментарии не отбрасываются, а запоминаются как пробелы, следующие непосредственно за лексемой. Далее происходит препроцессорная обработка текстов, заключающаяся в построении таблицы макросов, выполнении условной компиляции, включении файлов и замены макроподстановок.

После препроцессорной обработки нужен дополнительный лексический анализ, т к. в языке препроцессора языка C [2] есть несколько инструментов для создания лексем — это операции `#` и `##`. Их применение приводит к тому, что несколько лексем, полученных во время предварительного лексического анализа, объединяются в одну. При это тип новой лексемы зависит от типа её составляющих.

После дополнительного лексического анализа происходит синтаксический анализ. Во время синтаксического анализа из входной цепочки лексем выбираются последовательности лексем, удовлетворяющие одному из правил определения внешних деклараций. Внешними декларациями (external declaration в [2, 4] являются: определение прототипа функции, определение тела функции, определение глобальной переменной, определение `typedef`-имени, определение типа.

Некоторые внешние декларации могут комбинироваться, например: `struct M {...} A, *B, f(void);`. В этой декларации скомбинированы: определение типа, определение глобальной переменной, определение прототипа. Все декларации начинаются либо со служебного слова `typedef`, либо с декларации типа (в случае, когда тип опущен, считается что объявляемый тип — это `int`). Если декларация начинается с определения типа, то сначала происходит отделение и запоминание цепочки, соответствующей декларации типа (здесь `struct M {...}`).

Далее, для каждого идентификатора в цепочке A, *B, f(void) производится следующая операция: выбирается цепочка лексем, соответствующая одному идентификатору (выражение между запятыми), к ней присоединяется цепочка, соответствующая объявлению типа. Далее происходит обработка полученной объединенной цепочки. Так, в данном случае будут рассмотрены цепочки `struct M {...} A`, `struct M {...} *B`, `struct M {...} f(void)`. После этого каждая полученная цепочка разбирается в отдельности.

Если декларация начинается со слова `typedef`, то анализатор действует по предыдущей схеме с единственным исключением: получаемые объекты заносятся в таблицу типов, а не в таблицы переменных и функций. Для каждой внешней декларации запускается свой анализатор, который выполняет свои специфические действия.

Анализ типов. Во всех внешних декларациях участвуют типы. При объявлении прототипа функции или глобальной переменной в программе происходит декларация типа. Некоторые типы могут не иметь имени. Например: `struct {...} f(int x);` Тип, который возвращает функция не имеет имени.

Анализатор деклараций типов работает рекурсивно, определяя дополнительные типы (указатели, массивы, функции, константные типы) по мере необходимости. Поиск среди существующих типов осуществляется не по имени (типы могут не иметь имен, как отмечалось выше), а по форме конструирования из других типов.

Анализ прототипов функций. При обработке происходит (при необходимости) пополнение таблицы функций новым элементом. При этом имена параметров не запоминаются, в таблицу заносится только их тип. В поле определения тела функции выставляется константа, соответствующая несуществующему участку текста.

Анализ определений функций. Происходит (при необходимости) пополнение таблицы функций или модификация существующего элемента. В поля элемента заносится та же информация, что и для прототипа, запоминаются имена аргументов. Далее выделяется операторный блок — один большой составной оператор, ограниченный фигурными скобками (`{...}`), и происходит его рекурсивный разбор на составляющие простые операторы. Простыми операторами считаются: пустой оператор, оператор объявления переменной, арифметическое выражение, оператор условия, операторы цикла (`while`, `for`, `do`), оператор перехода, операторы меток, операторы `continue`, `break`, `return`, `switch`. Операторный граф получается путём связывания операторов по вложенности и по управлению.

Далее, для всех операторов, являющихся арифметическими выражениями, происходит их более детальное исследование: строится арифметическое дерево, как описано в [1, 3]. Также происходит анализ арифметических выражений, стоящих в операторе `return`. В построенном дереве листьями являются идентификаторы и константы, а узлами — арифметические операции. После этого происходит поиск идентификаторов в таблицах в следующем порядке: поиск в таблице локальных переменных, затем в таблице параметров функции, в таблице переменных и наконец, в таблице функций. Для

найденного объекта происходит пополнение таблиц использования: для самого объекта добавляется позиция использования; среди свойств оператора происходит пополнение таблицы ссылок на использованные объекты с указанием характера использования («чтение», «запись», «взятие адреса», «разыменование указателя» и «передача управления»). Передачей управления называется вызов функций через переменные-указатели, которые обычно передаются в качестве параметров в функции.

Анализ определений глобальных переменных и констант. Происходит пополнение соответствующей таблицы новым элементов. Обрабатывая исходный текст, анализатор сохраняет результаты в промежуточное ВП. После анализа всех исходных тестов исследуемого программного обеспечения происходит объединение промежуточных ВП. Этот процесс имеет прямую аналогию с работой редактора внешних связей — сборщика.

Кроме внутренних (встроенных) анализаторов, предусмотрена возможность использовать внешние анализаторы. Идея подключения внешних анализаторов заключается в том, что большинство трансляторов и компиляторов в процессе своей работы строят своё ВП, в общих чертах похожее на ВП, используемое в комплексе. Таким образом, если имеется возможность получить доступ к ВП внешнего транслятора, то при помощи специальной подсистемы-конвертера можно преобразовать ВП внешнего транслятора в ВП комплекса «Тест». Для анализа исходных текстов на языке C++ используется компилятор переднего плана от фирмы Interstron. Этот компилятор предоставляет доступ к своему внутреннему представлению через COM интерфейс. Построение ВП исходных текстов на языке C++ выглядит так: для каждого исходного файла запускается компилятор; затем, используя специальный модуль-переводчик, ВП компилятора преобразуется в промежуточное ВП комплекса «Тест»; далее, несколько промежуточных ВП объединяются в одно при помощи сборщика. Использование внешних анализаторов позволяет существенно расширить множество исследуемых языков.

Литература

[1] Ахо, Ульман. Теория синтаксического анализа, перевода и компиляции. Т. 2. Компиляция. М: Мир, 1978, 487 с.

[2] International standard ISO/IEC 9899:1999 (E) Programming languages — C, Second edition. Электронная версия. 550 с.

[3] Донован Дж. Системное программирование. М: Мир, 1975, 540 с.

[4] Керниган Б., Ритчи Д. Язык программирования Си. М: Финансы и статистика, 1992, 271 с.

Применение методов статического анализа для проверки свойств безопасности программ

Д. М. Русаков

Задача проверки свойств безопасности программного обеспечения относится к числу наиболее актуальных задач современного программирования. Эта задача возникает, в частности, при выявление недокументированных сценариев работы системы. В настоящее время для ее решения часто приходится вручную анализировать исходный код. Объясняется это, прежде всего, тем, что задача выявления многих свойств программных систем относится к числу алгоритмически неразрешимых проблем. Так, например ни одно нетривиальное свойство вычислений программ не допускает надежной автоматической проверки. Тем не менее, существуют методы, которые позволяют получать некоторые предварительные заключения о качестве и характере вычислений программ, и в ряде случаев эти заключения могут оказаться точными и окончательными.

Для выявление многих свойств программных систем применим статический анализ потоков данных. Под статическим анализом потоков данных понимается такой метод оценки поведения программы, при котором знания об интересующем нас свойстве могут быть получены на основе установления определенных зависимостей между отдельными компонентами программы (операторами,

переменными, процедурами, и т. п.) без привлечения результатов тестовых экспериментов. Известно (см. [2]), что в большинстве случаев получение точного решения задачи статического анализа также является алгоритмически неразрешимой проблемой. Однако для практических целей чаще всего можно ограничиться получением некоторых оценок этого точного решения, и это приводит к необходимости разработки эффективных аппроксимирующих алгоритмов статического анализа программ. Так, например, не существует алгоритма, который позволял бы получать точный ответ на запрос об инициализированности заданной переменной в заданной точке программы. Но при этом можно построить полуразрешающий алгоритм, который, в случае получения положительного ответа на указанный запрос, позволяет гарантировать, что заданная переменная в заданной точке программы при любом вычислении будет иметь предписанное ей значение.

Классическое решение задачи статического анализа потоков данных предполагает построение систем уравнений, описывающих потоки данных в каждой точке программы, и последующее решение построенной системы. Эффективность алгоритмов решения систем уравнений потоков данных во многом определяется как видом самих уравнений, так и способом их представления.

Для составления системы уравнений потоков данных, характеризующих исследуемое свойство поведения программы, вводится решетка абстрактных данных $\mathcal{L}$ конечной высоты. Элементы этой решетки могут быть истолкованы как знания об анализируемом свойстве, допустимые в рамках введенной абстракции.

$$P_i = f_i(g(P_{j_1}, P_{j_2}, \ldots, P_{j_n})), \tag{1}$$

где

- $P_m \in \mathcal{L}$ — состояние абстрактных данных (значение анализируемого свойства) после выполнения m-ого оператора,
- $j_1, j_2, \ldots, j_n$ — номера всех операторов, предшествующих по управлению i-ому оператору (или следующих за i-ым оператором в зависимости от анализируемого свойства),
- $g\colon \mathcal{L}^n \to \mathcal{L}$ — монотонная по каждому параметру функция, общая для всех операторов (как правило, это операция вычисления точной верхней или нижней грани на решетке $\mathcal{L}$),
- $f_i\colon \mathcal{L} \to \mathcal{L}$ — монотонная (в том же смысле, что и g) функция, обладающая свойством дистрибутивности:
$$f_i(g(P_{j_1}, P_{j_2}, \ldots, P_{j_n})) = g(f_i(P_{j_1}), f_i(P_{j_2}), \ldots, f_i(P_{j_n}))),$$
$j_1, j_2, \ldots, j_n \in \{1, 2\ldots., k\}$, $i = 1, 2, \ldots, k$,
- функция f_i строится индивидуально для каждого оператора программы с учетом его семантики,
- k — число операторов в программе.

Часто в каждой точке программы анализируются значения нескольких свойств одновременно. Например, инициализированность каждой из переменных программы. Поэтому P_m часто определяют как вектор свойств $P_m = (P_m^1, P_m^2, \ldots, P_m^t)$, где $P_m^i \in \mathcal{L}$, $t \geqslant 0$ - количество анализируемых свойств. В этом случае функции g и f имеют вид: $g\colon \mathcal{L}^{n \times t} \to \mathcal{L}^t$, $f\colon \mathcal{L}^t \to \mathcal{L}^t$.

Тогда справедлива

Теорема 1. *Система уравнений* (1) *всегда имеет решение. При этом наименьшее решение системы* (1) *можно вычислить при помощи итерационного алгоритма за конечное число шагов.*

Более подробное описание методов статического анализа потоков данных представлено в [2].

Эффективность алгоритмов статического анализа существенно зависит от способа представления данных, связанных с системой уравнений (1). Нами использовался метод, описанный в [1], где система представляется посредством размеченного графа потока управления программы. Графом потока управления принято называть граф, вершинами которого являются операторы программы, а дуги соответствуют возможности непосредственной передачи управления от одного оператора другому. Метод исходит из следующих предположений:

С1. Число анализируемых свойств ограниченно.

C2. Уравнения $P_i = f_i(P_j)$, где $i, j \in \{1, \ldots, k\}$, представимы в виде:

$$P_i^s = h(f_i^1(P_j^1), f_i^2(P_j^2), \ldots, f_i^t(P_t^2)),$$

где $s = 1, 2, \ldots, t$, $f_i^q \colon \mathcal{L} \to \mathcal{L}$, $q \in \{1, 2, \ldots, t\}$, $h \colon \mathcal{L}^t \to \mathcal{L}$ — общая для всего алгоритма функция.

C3. Уравнение $P_i = g(P_1, P_2, \ldots, P_n)$ представимо в виде

$$P_i^s = g^s(P_1^s, P_2^s, \ldots, P_n^s),$$

где $s = 1, 2 \ldots, t$.

Как видно, все три предположения почти всегда выполняются на практике.

Используя дистрибутивность рассматриваемых функций приведем каждое уравнение системы (1) к виду

$$P_i = g(f_i(P_{j_1}), f_i(P_{j_2}), \ldots, f_i(P_{j_n}))).$$

Тогда вся система уравнений может быть представлена в следующем виде:

$$\begin{aligned} P_i^s = g^s(h(f_i^1(P_{j_1}^1), f_i^2(P_{j_1}^2), \ldots, f_i^t(P_{j_1}^t))), \ldots, \\ h(f_i^1(P_{j_n}^1), f_i^2(P_{j_n}^2), \ldots, f_i^t(P_{j_n}^t)))), \end{aligned} \tag{2}$$

$s = 1, 2, \ldots, t$.

Функции g^s и h не зависят от отдельных операторов, содержащихся в программе, и определяются только анализируемым свойством и выбором абстрактных данных (решеткой $\mathcal{L}$). Рассмотрим следующий теоретико-графовый способ представления функций f_i^s. Для этого используются полные двудольные размеченные ориентированные графы специального вида. Каждая доля такого графа содержит s вершин; эти вершины упорядочены, и i-ая вершина каждой доли помечена элементом a_i, где $a_1, a_2, \ldots, a_n$ — все анализируемые свойства. Каждой дуге $a_p \to a_r$ приписана формула f_i^{pr}.

Тогда система уравнений (2) может быть приведена к следующему каноническому виду

$$\begin{aligned} P_i^s = g^s(h(f_i^{1j_1}(P_{j_1}^1), f_i^{2j_1}(P_{j_1}^2), \ldots, f_i^{tj_1}(P_{j_1}^t))), \ldots, \\ h(f_i^{1j_n}(P_{j_n}^1), f_i^{2j_n}(P_{j_n}^2), \ldots, f_i^{tj_n}(P_{j_n}^t)))). \end{aligned}$$

Построенные таким образом двудольные графы подставляется в граф потока управления следующим образом: каждая дуга в графе потока управления замещается соответствующим этой дуге двудольным графом.

Решение системы проводится в два этапа:

1. Для каждой процедуры и для каждого содержащегося в этой процедуре оператора строится двудольный граф, представляющий функции изменения анализируемых свойств.

2. На основе полученных функций вычисляются значения свойств для каждого оператора, зависящие от значений свойств на входе программы.

При реализации этого алгоритма возникают задачи выбора компактного и эффективного способа представления функций канонической системы уравнений, а также эффективного задания операций над указанными функциями.

Справедлива следующая теорема

Теорема 2. *Если функции g^s, f_i^{kj} монотонны и дистрибутивны, и при этом каждая функция f_i^{kj} может быть представлена структурой данных, размер которой ограничен некоторой константой, а время выполнения каждой операции над этими функциями также ограничено некоторой константой, то в случае выполнения условий C1–C3 алгоритм статического анализа потоков данных корректен и время его работы не превышает величины $O(ED^3)$, где E — количество дуг в графе потока управления, D — число анализируемых для каждого оператора свойств.*

Описанный метод анализа потоков данных был реализован в виде универсальной программы статического анализа, позволяющей единообразно проводить анализ различных свойств программ. Эта программа может быть использована в качестве ядра перспективной системы проверки свойств безопасности программ. Нами были также реализованы и опробованы на ряде примеров специализированные варианты указанной программы, позволяющие решать следующие пять задач:

1. *Анализ свойства инициализированности переменной.* Для каждого оператора s анализируемой программы вычисляется список переменных, которые *могут* быть инициализированы перед выполнением оператора s. Если некоторая переменная x не входит в такой список, то это означает, что перед началом любого выполнения оператора s переменная x не будет инициализирована.

2. *Анализ свойства переменной иметь постоянное значение.* Для каждого оператора s анализируемой программы вычисляется список переменных, каждая из которых имеет одно и то же фиксированное значение перед началом любого выполнения оператора s.

3. *Анализ свойства значения переменной иметь постоянный знак.* Для каждого оператора s анализируемой программы вычисляется список переменных, каждая из которых *может* иметь отрицательное значение перед началом любого выполнения оператора s. Если некоторая переменная x не содержится в таком списке, то это означает, что перед началом любого выполнения оператора s она *обязательно* имеет неотрицательное значение.

4. *Проверка использования переменных.* Для каждой переменной x проверяется, используется ли эта переменная в некотором операторе анализируемой программы. В том случае, если переменная используется в операторе s, строится одна из синтаксически допустимых трасс возможного вычисления, ведущая в оператор s.

5. *Анализ глобальных переменных.* Для каждой процедуры вычисляет список возможно используемых и возможно изменяемых ею глобальных переменных.

Первые три задачи являются классическими задачами статического анализа потоков данных, в то время как для решения последних двух задач существуют и более простые подходы не привлекающие рассмотренный нами аппарат статического анализа программ. Тем не менее оказалось, что использование готового ядра ускоряет создание анализирующей программы и при этом оставляет производительность на приемлемом уровне.

Анализатор свойств программ был реализован на языке программирования C++ и приспособлен для анализа программ на облегченном подмножестве языка C. Ядро представляет собой набор классов, среди которых имеются абстрактные, определение потомков которых приводит к конкретизации задачи. Разработанный нами анализатор был опробован на ряде простых C-программ. В таблице 1 приводятся результаты проведенных экспериментов.

	my.c	prog1.c	compress.c
Количество строк	25	308	1200
Количество процедур	3	17	39
Время работы «инициализированность переменных» (мин:сек:млс)	370	290	8:662
Время работы «константность значений переменных» (мин:сек:млс)	371	301	9:073
Время работы «отрицательность переменных» (мин:сек:млс)	411	250	4:927
Время работы «изменение и использование глобальных переменных» (м:с:млс)	330	1:572	8:42:321
Время работы «трассы использования переменных» (мин:сек:млс)	30	320	5:378

Таблица 1:

Литература

[1] Reps T. Program analysis via graph reachability // In: Proceedings of the International Symposium on Logic Programming. 1997. P. 5–19.

[2] Neilson F., Nielson H., Hankin C. // Principles of Program Analysis. Berlin-Heidelberg: Springer-Verlag, 1999.

О противодействии некоторым алгоритмам статического анализа программ[14]

К. С. Иванов, В. А. Захаров

Задача обфускации компьютерных программ заключается в разработке таких эффективных алгоритмов преобразования программ, в результате применения которых к любой программе π из заданного класса будет получена новая программа π', удовлетворяющая следующим требованиям.

1. *Функциональность.* Программа π' вычисляет ту же самую функцию, что и исходная программа π.
2. *Эффективность.* Качественные характеристики программы π' (быстродействие, объем используемой памяти, размер программы, и т. п.) «не очень значительно» ухудшаются по сравнению с аналогичными характеристиками программы π.
3. *Стойкость.* Задача распознавания некоторых выделенных свойств программы π на основе текста программы π' не может быть решена с использованием определенных алгоритмических средств за разумное время.

Требование стойкости отражает пожелание автора программы защитить те или иные секреты (особенности устройства алгоритма и структур данных, используемые в алгоритме параметры и константы, и др.), которые проявляются в тексте спроектированной программы, но не подлежат публичному разглашению. Обнаружение противником секретных свойств программы и составляет угрозу, для предотвращения которой применяются методы обфускации программ. Задача обфускации компьютерных программ с целью затруднения извлечения из текста программы полезной информации об устройстве заложенного в ней алгоритма и предотвращения внесения несанкционированных изменений в программу была поставлена в работе [4]. В этой же статье был выделен ряд эквивалентных преобразований, полезных для проведения обфускации программ, проведена классификация этих преобразований, а также предложены некоторые неформальные критерии, позволяющие оценить качество обфускирующих преобразований. Последующие результаты исследований в этом направлении [3, 5, 7, 8] носили в основном эвристический характер, ввиду отсутствия строгой формулировки требования стойкости обфускации. Первое формальное определение стойкости обфускации (стойкость «виртуального черного ящика») было предложено в [2]. Обфускатор обладает стойкостью «виртуального черного ящика», если произвольный вероятностный алгоритм достигает за полиномиальное время не большего успеха в анализе текста обфускированной программы, нежели некоторая система, проводящая тестовые эксперименты с программой в режиме «черного ящика» (без доступа к тексту тестируемой программы). В [2] было установлено, что универсальных стойких обфускаторов не существует.

Однако для многих практических приложений требование стойкости «виртуального черного ящика» может быть значительно ослаблено. Для раскрытия секретов, содержащихся в тексте программы, противник может применять различные методы, включая тестирование, статический анализ программ, декомпозицию программ и др. Решение некоторых задач анализа программ может быть до определенной степени автоматизировано. Имеется большой арсенал алгоритмов, используемых для верификации и оптимизации программ, извлекающих из программы на этапе компиляции полезную информацию о свойствах и взаимосвязях между различными ее компонентами. В таком случае стойкость обфускирующих преобразований можно оценивать в зависимости от того, в какой мере эти преобразования затрудняют применение или ухудшают результаты работы известных алгоритмов статического анализ программ. В настоящей работе мы предлагаем один из возможных подходов к разработке и оценке способности обфускирующих преобразований противодействовать алгоритмам

[14]Работа выполнена при поддержке гранта РФФИ 03-01-00880.

статического анализа программ. В основу этого подхода положена следующая идея. Предположим, что рассматривается алгоритм статического анализа программ S применительно к классу программ P. В классе P выделяется подкласс программ P_0, анализ которых при помощи алгоритма S дает наименее точный результат. Программы из подкласса P_0 будем называть *непрозрачным для алгоритма* S. Тогда для эффективного противодействия алгоритму S достаточно решить следующие три задачи:

1. Убедиться в том, что для любой программы из класса P существует эквивалентная ей непрозрачная программа.
2. Построить систему эквивалентных преобразований программ, позволяющую преобразовать всякую программу из класса P в эквивалентную непрозрачную программу.
3. Разработать эффективную процедуру эквивалентного преобразования программ из класса P в подкласс непрозрачных программ.

Предложенный подход к повышению стойкости программ по отношению к средствам декомпиляции был опробован для двух типов алгоритмов статического анализа программ: алгоритмов оценки диапазонов переменных-указателей и алгоритмов построения сечения.

Алгоритмы оценки диапазонов значений переменных-указателей призваны вычислять для каждой точки L программы (или для программы целиком, в этом случае параметр L опускается) и для каждой переменной-указателя X множество $R(X, L)$ всех тех переменных (адресов) программы, на которые может ссылаться переменная-указатель X при некотором прохождении вычисления программы через точку L. Поскольку для некоторых программ множество $R(X, L)$ оказывается нерекурсивным, на практике приходится ограничиваться вычислением верхней оценки $R'(X, L) \supseteq R(X, L)$. Наименее точный результат анализа получается в том случае, когда множество $R'(X, L)$, вычисленное алгоритмом S оценки диапазонов переменных-указателей, включает все переменные, используемые в программе. В таком случае будем говорить, что анализируемая программа *непрозрачна для алгоритма* S.

В качестве алгоритма оценки диапазонов переменных-указателей нами был рассмотрен алгоритм Андерсена [1], который широко используется в современных системах оптимизации и верификации программ. Для алгоритма Андерсена был установлена справедливость следующих утверждений.

Утверждение 1. *Для каждой программы π существует эквивалентная ей программа π', являющаяся непрозрачной для алгоритма Андерсена оценки диапазона значений переменных-указателей.*

Утверждение 2. *Существует система эквивалентных преобразований программ, позволяющая перевести каждую программу π в эквивалентную ей непрозрачную программу π'.*

Утверждение 3. *Существует эффективная процедура приведения всякой программы к непрозрачному для данного алгоритма виду, и при этом сложность по времени указанной процедуры приведения квадратична относительно числа переменных и линейна относительно размера программы.*

Главная слабость рассматриваемого алгоритма состоит в нечувствительности к потоку управления. Рассмотрим две операции присваивания адреса:

$$p = \&a;$$
$$q = \&a;$$

После анализа мы получим, что указатели p и q являются синонимами независимо от количества и состава операторов, располагающихся между ними.

В основу предложенной нами системы эквивалентных преобразований программ, противодействующих алгоритму Андерсена, положен следующий замысел: используя операторы присваивания и разыменования, а также дополнительные переменные-указатели сделать все переменные-указатели одного уровня вложенности синонимами друг друга. Процедура преобразования состоит из пяти этапов:

- анализ программы;
- исследование и оптимизация данных анализа;
- подготовка дополнительных переменных-указателей;

- создание операторов для новых переменных;
- внедрение новых операторов в программу, с помощью методов локального сохранения значений.

Алгоритмы построения сечений призваны вычислять для каждой точки L программы и для каждой переменной X фрагмент $F(X, L)$ программы, включающий все операторы, от которых зависит значение переменной X в точке L. Наименее точный результат анализа получается в том случае, когда множество $F(X, L)$, вычисленное алгоритмом S построения сечений, состоит из всех операторов программы, предшествующих точке L. В таком случае мы будем говорить, что анализируемая программа *непрозрачна для алгоритма S построения сечений.*

Нами был рассмотрен алгоритм построения сечения Репса-Хорвитц [6], используемый во многих исследовательских и коммерческих системах анализа программ. Применительно к алгоритму Репса-Хорвитц было установлена справедливость следующих утверждений.

Утверждение 4. *Для каждой программы π существует эквивалентная ей программа π', являющаяся непрозрачной для алгоритма Репса-Хорвитц построения сечений программ.*

Утверждение 5. *Существует система эквивалентных преобразований программ, позволяющая перевести каждую программу π в эквивалентную ей непрозрачную программу π'.*

Утверждение 6. *Существует эффективная процедура приведения всякой программы к непрозрачному для данного алгоритма виду.*

Основная идея заключается в введении дополнительных зависимостей (прямых или косвенных) в каждую инструкцию программы от всех предыдущих. Программа просматривается последовательно, определяется тип каждой инструкции. В зависимости от этого типа выполняются маскирующие преобразования, на основе прибавления константы. Оператор присваивания $y = f(x)$ преобразуется в оператор присваивания $y = f(x) + g(z) - c$, где выражение $g(z)$ выбирается так, чтобы в нем содержались все переменные предыдущих операторов, но значение этого выражения всегда тождественно равнялось c. После такого преобразования каждый оператор программы будут зависеть от всех предшествующих ему операторов.

Полученные результаты свидетельствуют о том, что имеется практическая возможность построения обфускаторов, способных противодействовать некоторым стандартным инструментальным средствам анализа программ. В дальнейшем нами планируется исследовать более широкий круг таких инструментальных средств и попытаться разработать методы противодействия наиболее точным и эффективным алгоритмам анализа программ.

Литература

[1] Andersen L. O. Program analysis and specialization for C programming language. DIKU, Univ. of Copenhagen, May 1994.

[2] Barak B., Goldreich O., Impagliazzo R., Rudich S., Sahai A., Vedhan S., Yang K. On the (Im)possibility of obfuscating programs. CRYPTO'01 - Advances in Cryptology, Lecture Notes in Computer Science, v. 2139, 2001, p. 1–18.

[3] Chow S., Gu Y., Johnson H., Zakharov V. An approach to the obfuscation of control flow of sequential computer programs. Information Security Conference, Lecture Notes in Computer Science, v. 2200, 2001, p. 144–156.

[4] Collberg C., Thomborson C., Low D. A taxonomy of obfuscating transformations. Tech. Report N 148, Dept. of Computer Science, Univ. of Auckland, 1997.

[5] Collberg C., Thomborson C., Low D. Manufacturing cheap, resilient and stealthy opaque constructs. Symposium on Principles of Programming Languages, 1998, p. 184–196.

[6] Horwitz S., Reps T., Binkley D. Interprocedural slicing using dependence graphs. ACM Transactions on Programming Languages and Systems, v. 12, No 1, p. 26–60.

[7] Wang C., Hill J., Knight J. Davidson J. Software tamper resistance: obstructing static analysis of programs. Tech. Report N 12, Dep. Of Comp. Sci., Univ. of Virginia, 2000.

[8] Wroblewski G. General method of program code obfuscation. in: Proceedings of the International Conference on Software Engineering Research and Practice (SERP), 2002, p. 153–159.

Протоколирование и фильтрация системных вызовов в ядре ОС Linux

С. А. Ахманов

1 Введение

Практически в любой современной операционной системе (ОС) существуют стандартные средства протоколирования системных вызовов. Их основная задача — обеспечить для системных администраторов и разработчиков возможность отслеживать взаимодействия прикладных программ и ядра ОС. В некоммерческих операционных системах семейства Unix такими средствами являются strace и ktrace для Linux и FreeBSD соответственно. Стандартных средств протоколирования системных вызовов во многих случаях бывает недостаточно для выполнения ряда задач, что вызывает необходимость разработки новых инструментов. Для ОС Linux, ядро которой рассматривается в настоящей публикации, существует ряд альтернативных программных средств для решения этой задачи. Они имеют различное назначение и функциональность. Некоторые из них будут рассмотрены ниже.

Фильтр системных вызовов — это набор программных средств (на уровне ядра и на пользовательском уровне), позволяющий системному администратору выборочно блокировать системные вызовы. Фильтр является дополнительным механизмом безопасности на уровне ядра операционной системы. Применение фильтров различных типов позволяет обеспечить защиту системы от ряда атак, для которых применение других средств защиты либо затруднительно, либо вообще невозможно.

Задачи выборочного протоколирования системных вызовов и выборочной блокировки системных вызовов тесно связаны между собой. Схемы их реализации схожи. Поэтому мы рассматриваем эти задачи вместе как единую задачу разработки дополнительных средств обеспечения безопасности на уровне ядра операционной системы.

Основными результатами данной работы являются реализованный и протестированный фильтр — набор программных средств на уровне ядра и пользовательском уровне, а также разработанная архитектура фильтрующего механизма, которая применима как для реализации фильтра системных вызовов, так и для решения задачи их протоколирования. В рамах данной архитектуры реализована двухуровневая система протоколирования системных вызовов. Кроме того, представлены результаты тестирования производительности (с применением фильтра и без него) для нескольких типичных компьютерных систем.

1.1 Сферы применения фильтра системных вызовов

Приведем примеры конкретных задач, когда необходимо протоколирование системных вызовов. Для решения некоторых из них стандартных средств протоколирования, таких как strace и ktrace, оказывается недостаточно.

1. Протоколирование доступа к критически важным для безопасности системы файлам.

Рассмотрим эту задачу на примере файла `/etc/shadow`, в котором хранятся затенённые хэш-значения паролей пользователей в большинстве Unix-систем. Доступ к этому файлу имеют все процессы, обладающие привилегиями суперпользователя. В качестве дополнительной меры безопасности имеет смысл протоколировать доступ к этому файлу и реагировать в случае, если доступ к нему осуществляется процессом, не связанным с аутентификацией пользователей системы.

2. Отладка программного обеспечения.

3. Анализ программного обеспечения с отсутствующим исходным кодом.

Типичные задачи, для решения которых необходим фильтр системных вызовов, можно охарактеризовать следующим образом.

1. Ограничение роли пользователя.

Для некоторых пользователей системы, выполняющих ограниченный круг задач, например, операторов, вводящих данные в информационную систему вручную или при помощи промышленного сканера, имеет смысл ограничить набор системных вызовов, который необходим для работы их приложений.

В последнее время не редко появляются сообщения об обнаружении уязвимостей в ядрах операционных систем семейства Unix, дающих возможность любому пользователю, запустившему определенный код получить права суперпользователя. В некоторых случаях, например, в случае недавно обнаруженной уязвимости в ядре Linux до версии 2.4.18 в подсистеме pthread, правильно настроенный фильтр для непривилегированного пользователя позволит сохранить достаточный уровень безопасности системы до обновления ядра.

Аналогично предыдущему случаю, имеет смысл ограничить набор системных вызовов для процессов, запущенных от имени «служебных» учетных записей, таких как mail, ntp, squid, apache. Эти учетные записи предназначены для того, чтобы, в целях усиления безопасности, запускать соответствующие сетевые сервисы (почтовый сервер, time-сервер, кэширующий прокси-сервер, web-сервер) с ограниченными привилегиями. Такие меры принимаются в рамках работ по формированию модели эшелонированной защиты компьютерной системы. Таким образом, если будет обнаружена ошибка в коде сетевого сервиса, а атакующий воспользуется ею и получит доступ в систему с привилегиями данного процесса, то нанесение значительного ущерба будет маловероятно из-за низкого уровня привилегий данного процесса в системе. Атака на следующий эшелон защиты потребует дополнительных затрат времени, которое, как правило, бывает достаточно системам активного аудита для адекватной реакции на первую атаку.

Применение фильтра системных вызовов в такой схеме создает еще один эшелон обороны, усиливая тем самым общий уровень безопасности системы.

2. Запрет редко используемых системных вызовов.

Редко используемые системные вызовы, например mount и umount, в некоторых случаях имеет смысл полностью запретить. Как только система загрузится и все необходимые файловые системы будут подключены, запрет mount позволит закрыть одну из потенциальных уязвимостей системы, а запрет umount закроет для атакующего возможность дестабилизировать работу системы.

Изложенные выше области применения фильтра тесно согласуются с Общими Критериями Оценки Безопасности Информационных Технологий [2].

Положение `FIA_USB` Общих Критериев («User-subject binding») согласуется с задачей ограничения «роли» пользователя с помощью фильтра системных вызовов.

Положение `FAU_GEN` Общих Критериев («Security audit data generation») согласуется с задачей протоколирования системных вызовов.

Помимо изложенных выше сфер применения фильтра, существуют и перспективные сферы его применения. Приведем пример системы активного аудита, использующей систему протоколирования системных вызовов для получения оперативной информации.

Рассмотрим типичный почтовый сервер, например sendmail. Демон (неинтерактивный, не связанный с консолью, процесс) sendmail и все его процессы-потомки выполняют вполне ограниченный круг задач. Естественно, набор и семантика системных вызовов, которые они используют, ограничены. Если, например, данный почтовый сервер будет подвержен успешной атаке типа «переполнение буфера», ее результатом будет, скорее всего, запущенная атакующим пользовательская оболочка (shell). Допустим, атакующий попытается воспользоваться полученной оболочкой, например, будет просматривать какие-то файлы и каталоги. Набор системных вызовов, используемых процессом sendmail и его процессами-потомками, изменится.

Таким образом может быть построена система активного аудита, основанная на выборочном протоколировании системных вызовов. В данном случае достаточно предварительно собрать статистику

системных вызовов в случае типичного поведения, а далее, в случае не типичного поведения, любо остановить почтовый сервер и все его процессы-потомки, либо с помощью фильтра системных вызовов запретить им все системные вызовы до вмешательства администратора. Кроме того, последующий анализ результатов протоколирования системных вызовов позволит точно найти место в исходных текстах кода почтового сервера, в котором была найдена уязвимость.

На выборочном протоколировании системных вызовов можно построить не только статистическую систему активного аудита. Здесь возможны различные подходы. В частности, типичное и нетипичное поведение можно различать с помощью нейронной сети.

1.2 Известные на сегодняшний день разработки в данной области

Начнем со стандартных средств протоколирования системных вызовов. Для ОС Linux это strace. Основное назначение strace — протоколирование системных вызовов и сигналов для данного процесса (или поддерева процессов) с пользовательского уровня. Strace применяется, в основном, для отладки программного обеспечения (ПО), а также анализа ПО с отсутствующем исходным кодом. Для нужд администрирования применение strace неудобно, а во многих случаях вообще невозможно.

Strace реализован в виде пары: прикладной программы strace и системного вызова ptrace. Как правило, в качестве аргумента командной строки strace получает имя исполняемого файла. Результаты протоколирования (номера, имена, аргументы и возвращаемые значения системных вызовов, номера и имена событий) strace выводит в текстовом виде в stderr (стандартный небуферизуемый вывод для сообщений об ошибках), либо в заданный файл. При этом у пользователя нет возможности отобрать вызовы или события. Протоколируются все события и вызовы, далее их можно отобрать только анализируя результат. По этой причине strace невозможно применять для таких задач как отслеживание одного системного вызова из тысяч (например, выявить попытку несанкционированного удаления файла в течение одного дня).

Стандартное средство протоколирования системных вызовов для ОС FreeBSD (а также, OpenBSD, NetBSD) — ktrace. Функциональность ktrace аналогична функциональности strace. Основная деталь реализации, отличающая ktrace от strace — разделение утилиты пользовательского уровня на две: ktrace и kdump. Первая утилита аналогична strace, но результаты протоколирования она записывает в бинарный файл специального формата. Объемы таких dmp-файлов значительно меньше результатов, возвращаемых strace. Кроме того, процесс генерирования такого файла эффективнее, чем процесс генерирования текстового файла. Далее полученный dmp-файл можно расшифровать утилитой kdump, при этом, естественно, можно отобрать, используя механизм регулярных выражений, только интересующую пользователя информацию.

Такое разделение функциональности позволяет немного расширить сферу применения стандартных механизмов протоколирования системных вызовов, но, тем не менее, ktrace не является средством выборочного протоколирования.

Стандартные средства протоколирования системных вызовов не являются средствами выборочного протоколирования. Это сильно сужает область их применения. Кроме того, при протоколировании стандартными средствами производительность падает многократно. Это связано с необходимостью передачи относительно больших объемов данных с уровня ядра на пользовательский уровень. Таким образом, основная задача альтернативных программных средств — во-первых, обеспечить возможность отбора вызовов на уровне ядра, во-вторых — минимизировать затраты ресурсов на решение поставленной задачи.

На сегодняшний день ни Linux, ни FreeBSD (OpenBSD, NetBSD) не имеют стандартных средств выборочной блокировки системных вызовов. При этом, как для задачи протоколирования системных вызовов, так и для задачи выборочной блокировки системных вызовов существуют альтернативные программные средства. Их назначение и архитектура различны. Далее мы рассмотрим ряд наиболее известных продуктов.

Один из наиболее известных альтернативных продуктов — System Call Tracker, разработка Haifa Linux Club [4]. Его первоначальное основное предназначение — выборочное протоколирование системных вызовов. Авторы предложили следующие примеры задач, для решения которых предназначен их продукт.

1. Допустим, некоторый файл в системе периодически удаляется по неизвестным причинам и

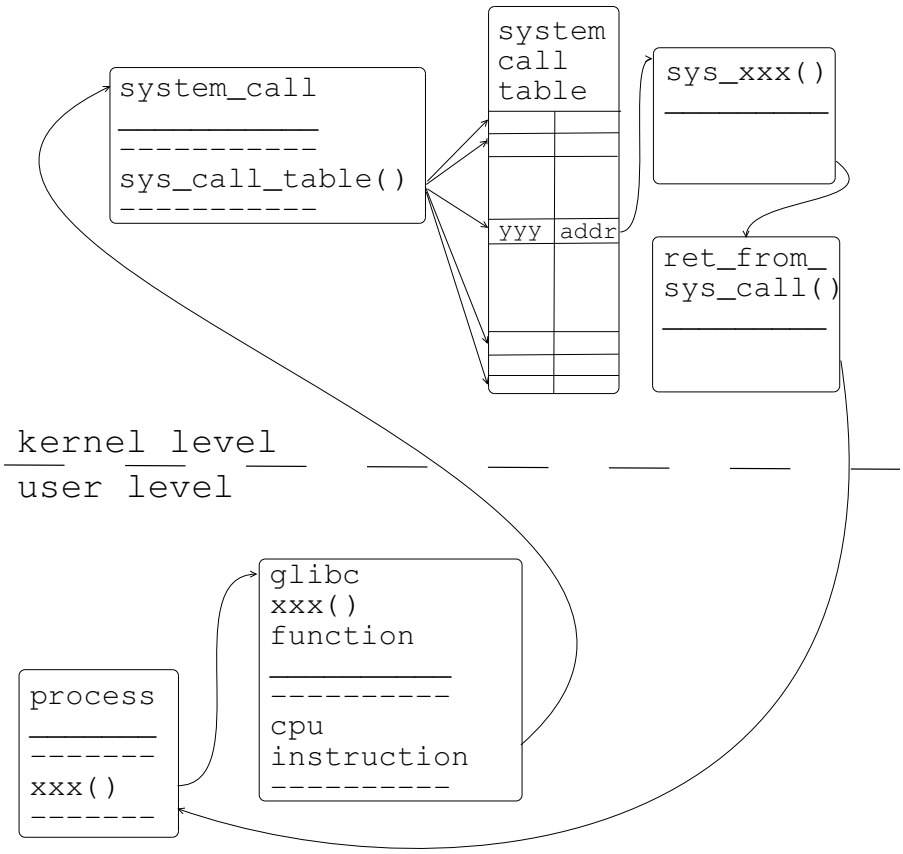


Рис. 1: Схема работы системного вызова.

требуется выяснить, какой процесс это делает. Авторы System Call Tracker предлагают следующее решение: отследить вызов `unlink()` с параметром — именем данного файла.

2. Некоторый процесс, по неизвестным причинам, получает сигнал `SIGTERM` и завершается. Отследить, откуда приходит сигнал `SIGTERM` можно протоколируя вызов `kill()`.

3. Права доступа к некоторому объекту изменяются и требуется отследить, какой процесс это делает. Протоколируя `chmod()` можно найти источник проблемы.

Изначально System Call Tracker разрабатывался для выборочного протоколирования системных вызовов, затем в продукт была добавлена функция фильтра. В дальнейшие планы авторов входит реализация возможности «подмены» системных вызовов, а именно — изменение возвращаемого значения в соответствии с правилами.

2 Подходы к решению. Выбор архитектуры.

Рассматривая архитектуру фильтров системных вызовов мы можем без потери общности говорить только о фильтре, не затрагивая задачу протоколирования. Из изложенного выше описания различных программных средств видно, что в большинстве случаев для обеих задач используется один и тот же, либо аналогичный код, различны только результаты работы.

Итак, на рисунке 1 изображена схема работы системного вызова в наиболее распространенном случае на примере вызова с именем `xxx()` и номером `yyy`. В рамках этой схемы видны возможные варианты для встраивания фильтра и места применения правил.

Стандартные средства протоколирования (strace, ktrace) реализованы в виде ветви кода после точки начала исполнения кода ядра. Стандартный код перехода к функции системного вызова по таблице векторов заменяется на аналогичный с функцией протоколирования (трассировки). При этом

сама таблица векторов системных вызовов остается неизменной, отличается только код ядра, выполняемый до и после вызова.

Такая архитектура удачна с точки зрения производительности. Если механизм трассировки выключен, потери производительности равны нулю, соответствующая ветвь кода вообще не выполняется. Для встроенных средств протоколирования, основным назначением которых является отладка программного обеспечения, это наиболее удачный вариант.

Как видно из рисунка 1, следующим возможным способом реализации фильтра является частичное или полное изменение таблицы векторов системных вызовов. Такой подход имеет одно очень существенное преимущество. Оно обусловлено тем, что в ядре Linux существует возможность изменять таблицу векторов системных вызовов из внешнего модуля ядра. Большинство альтернативных средств протоколирования и фильтрации используют именно такой подход.

Здесь возможны несколько вариантов. Вектора фильтруемых системных вызовов можно заменить на адрес фильтрующей функции, которая будет одна для всех вызовов. Далее, после применения правил, фильтрующая функция должна будет осуществить переход к основному коду системного вызова согласно сохраненной заранее таблице векторов системных вызовов. Этот вариант удачен, но потери производительности при большом наборе часто используемых фильтруемых системных вызовов здесь неизбежны. Такой механизм реализован в проекте Overloader [3].

Другой вариант — заменить вектора фильтруемых системных вызовов на адреса фильтрующих функций, причем для каждого системного вызова заводится отдельная фильтрующая функция. Такой подход удобен для реализации специальных фильтров системных вызовов, например, ориентированных только на вызовы, связанные с файловой системой. Для разработки универсального фильтра, использующего этот подход, необходимо применение средств автоматического генерирования кода. Такой механизм реализован в проекте System Call Tracker [4].

Безусловно, основная составная часть архитектуры фильтра системных вызовов — это фильтрующий механизм. Но помимо фильтрующего механизма есть и другие вспомогательные компоненты, архитектура которых должна быть определена на этапе проектирования в соответствии с назначением фильтра.

Метод передачи правил фильтра в память ядра. Здесь существует несколько вариантов. Процесс передачи правил должен инициироваться со стороны утилит пользовательского уровня, поэтому выбор средств передачи ограничен.

Файловая система `/proc` — набор переменных, находящихся в памяти ядра, доступных с пользовательского уровня в виде файловой системы — является достаточно удачным механизмом для решения данной задачи. Но на сегодняшний день не существует известных решений поставленной задачи при помощи `/proc`. Это вызвано тем, что наличие файловой системы `/proc` в Linux является необязательным. Данный механизм присутствует не в любой Unix-системе.

Авторы ядра ОС Linux рекомендуют авторам драйверов устройств выносить код драйвера во внешний модуль ядра, а для передачи параметров драйверу использовать системные вызовы `ioctl()` и `sysctl()`. Некоторые фильтры системных вызовов построены, в этом отношении, аналогично драйверу устройства.

И, наконец, автор фильтра может создать свои специальные системные вызовы для передачи правил фильтру. Такой способ не рекомендуется авторами ядра, однако он удобен и эффективен для разработчика.

При проектировании фильтра необходимо уделить должное внимание логике правил фильтра, способу хранения правил в памяти ядра, их синтаксису на пользовательском уровне и набору переменных, которые могут в них использоваться. В этом отношении каждый фильтр уникален. Приведем список части переменных, которыми располагает фильтр в момент применения правил (принятия решения):

- номер (и имя) запрашиваемого системного вызова;
- вся информация о процессе, из которого запрашивается вызов (содержимое структуры процесса):
  - pid,
  - ppid,
  - uid,
  - gid,

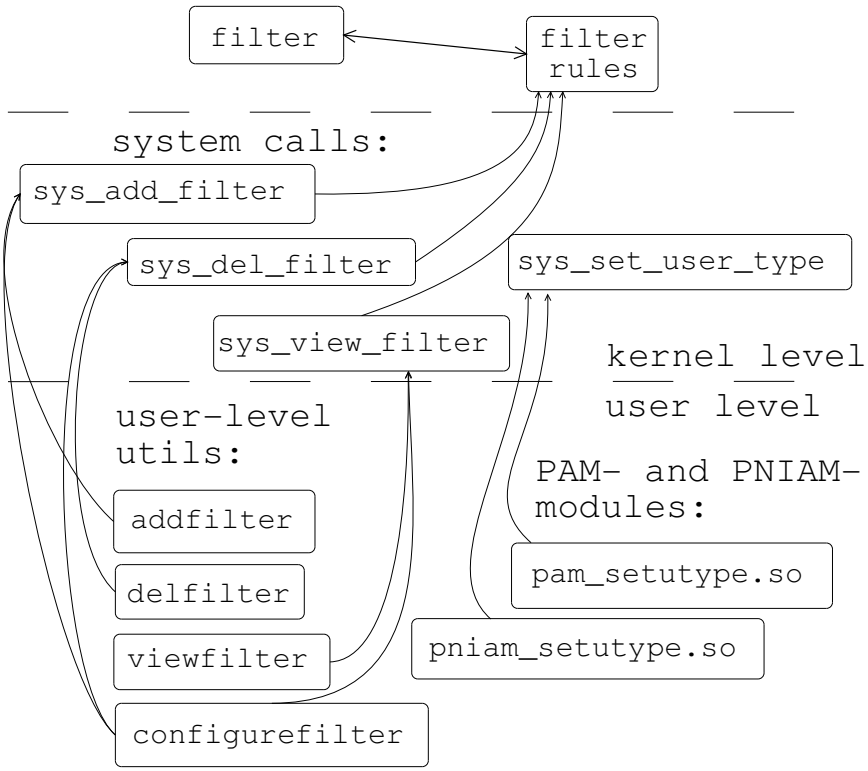


Рис. 2: Архитектура и компоненты фильтра.

  - информация о предках и потомках процесса в дереве,
  - номер процессора в SMP-системе,
  - ограничения (rlimits) на ресурсы, введенные для данного процесса
  - и т. д;
- значения параметров, передаваемых вызову.

3 Реализация фильтра

Как уже отмечалось, основным результатом данной работы является реализованный фильтр системных вызовов для ядра ОС Linux [6]. Рассмотрим его более подробно.

Еще раз остановимся на выбранной нами архитектуре. Схематически она изображена на рисунке 2.

Сначала об архитектуре фильтрующего механизма. В ядре Linux выполнение системного вызова начинается с функции `system_call`, реализованной на языке Ассемблер и находящейся в архитектурно-зависимой части кода ядра (файл `arch/i386/kernel/entry.S` исходных тестов ядра [6]). Эта функция выполняет проверки и осуществляет переход к коду системного вызова посредством таблицы векторов системных вызовов. После того, как системный вызов закончил работу, выполняется функция `ret_from_sys_call`, которая возвращает результат работы системного вызова, вызвавшему его процессу.

Проверка правил фильтра была встроена непосредственно в функцию `system_call`, при этом в случае необходимости заблокировать системный вызов `system_call` не передаст управление системному вызову согласно таблице векторов, а установит значение кода ошибки и вызовет `ret_from_sys_call`.

В рамках этой схемы правила протоколирования применяются дважды «до» и «после».

- Правила протоколирования «до» применяются в функции `system_call`. Это, в частности, необходимо чтобы протоколировать попытки вызова заблокированных вызовов.

- Правила протоколирования «после» применяются в функции `ret_from_sys_call`. Это позволяет протоколировать не только имя, номер и семантику вызова, но и результат — возвращаемое значение вызова.

Функция разбора правил фильтра также расположена в архитектурно-зависимой части кода ядра.

Передача правил с пользовательского уровня на уровень ядра осуществляется посредством трех специальных системных вызовов:

- добавить правило фильтра,
- удалить правило фильтра,
- просмотреть правило фильтра.

В первой реализации фильтра при формулировании правил мы оперировали только номером системного вызова, и специальным параметром, связанным с пользователем, от имени которого работает процесс, инициировавший данный вызов. Результатом, в случае если вызов удовлетворяет правилу, может быть либо блокировка вызова (вызов не выполнится, а процессу пользовательского уровня будет возвращён код ошибки), либо разрешение вызова.

Использовать uid (идентификатор пользователя) в качестве параметра, связанного с пользователем не эффективно, поэтому мы ввели свою переменную — `user_type`, тип пользователя. Посредством специального системного вызова тип пользователя устанавливается при входе в систему в процессе авторизации. Эта возможность легко встраивается в любую современную модульную систему аутентификации и авторизации, такую как, например, PAM [10] или PNIAM [11].

Фильтр реализован как «заплатка» для ядра Linux. Описанная реализация доступна для ядер версий 2.4.18 и 2.4.22. Часть параметров, необходимых для работы фильтра вынесена в конфигурацию ядра. На этапе конфигурирования ядра с наложенной заплаткой пользователь может включить или выключить фильтр, задать максимальное значение параметра `user_type`, включить режим отладки фильтра, в этом случае в протокол работы ядра будет заноситься подробная информация о работе фильтра.

Для управления фильтром с пользовательского уровня в рамка данной работы были разработаны четыре пользовательские утилиты. Три из них являются интерфейсами к системным вызовам, которые управляют фильтром: создать, удалить, просмотреть правило фильтра. Четвертая позволяет загружать несколько правил сразу из конфигурационного файла и, при необходимости, предварительно удалять все правила из фильтра.

Для инициализации фильтра для конкретного пользователя при его авторизации в системе были разработаны модуль `pniam_filter.so` для модульной системы аутентификации и авторизации PNIAM [11] и сценарии, конфигурирующие фильтр в типичных случаях: для суперпользователя или администратора, для обычного пользователя системы, для пользователя с ограниченными привилегиями в системе.

4 Влияние фильтра на производительность системы

Основной результат данной работы — это реализованный фильтр системных вызовов для ядра ОС Linux с достаточно эффективной архитектурой. Естественно в рамках такой работы получить реальные результаты измерения производительности.

Любой фильтр системных вызовов снижает общую производительность системы. Измерить, насколько это снижение влияет на работу, можно с помощью самого распространенного общего теста производительности для Unix-систем — Unixbench [12].

С помощью Unixbench 4.1.0 были получены достаточно правдоподобные результаты. Тест запускался сериями на несколько часов подряд ночью на системе, отключенной от компьютерной сети с минимальным количеством параллельно запущенных процессов. Одна серия испытаний проводилась с ядром версии 2.4.18, вторая — с тем же ядром с той же конфигурацией, но с встроенным фильтром. После подведения итогов мы получили результат средней потери общей производительности системы (FINAL SCORE по UnixBench) — 0,016. Т. е. потеря производительности составила в среднем 1,6%. Этот показатель можно оценить как незначительную потерю.

С другой стороны, представляет интерес получить результаты измерения производительности именно подсистемы системных вызовов ядра операционной системы. Для того, чтобы результаты теста были близки к реальности, мы собрали некоторую статистическую информацию о том, какие системные вызовы используются наиболее часто в различных компьютерных системах.

Приведем результаты сбора статистики системных вызовов для типичных случаев. При компилировании больших объемов кода (например, полной компиляции ядра) статистика системных вызовов показана в таблице 1, а для системы, находящейся в состоянии ожидания — в таблице 2.

номер	имя вызова	доля от общего количества вызовов
3	read	0.113337
5	open	0.114593
6	close	0.112951
45	brk	0.133561
192	mmap2	0.236100
197	fstat54	0.108575
	остальные вызовы	0.180883

Таблица 1:

номер	имя вызова	доля от общего количества вызовов
3	read	0.276808
4	write	0.068759
54	ioctl	0.103069
78	gettimeofday	0.161954
142	_newselect	0.226967
146	writev	0.068696
	остальные вызовы	0.093747

Таблица 2:

После анализа различных результатов сбора статистики системных вызовов, в рамках данной работы были реализованы несколько вариантов теста производительности подсистемы системных вызовов ядра. Набор вызовов, используемых в замерах был составлен по результатам сбора статистики.

В зависимости от набора вызовов, соответствующего типичным компьютерным системам, результат измерений падения производительности подсистемы системных вызовов ядра составил 8–9%. Результат аналогичного теста для стандартного средства протоколирования системных вызовов — strace — составил 98%. Таким образом, при применении стандартных средств этот показатель падает в 50 раз.

Таким образом, опираясь на первоначально полученный результат — 1,6%, можно утверждать, что одна из целей настоящей работы достигнута — разработан фильтр системных вызовов, лишь незначительно влияющий на производительность системы.

5 Заключение

В настоящей публикации представлена архитектура фильтра системных вызовов и его реализация. На основе анализа многочисленных результатов на данном направлении есть основание утверждать, что других реализаций в рамках выбранного подхода не существует.

Несмотря на то, что функциональность фильтра, разработанного в рамках данной работы ограничена, общая архитектура позволяет на аналогичных принципах решить задачи выборочного протоколирования системных вызовов, выборочной подмены системных вызовов, и, возможно, разработать соответствующую компоненту системы активного аудита с применением этих средств.

К основным достоинствам рассматриваемого в настоящей работе подхода можно отнести следующие. В отличии от стандартных средств протоколирования, в выбранной нами архитектуре правила фильтра хранятся в области памяти ядра, что эффективно с точки зрения производительности. С другой стороны, большая часть альтернативных средств выборочного протоколирования частично

или полностью заменяет таблицу векторов системных вызовов. Этот подход удобен для выноса фильтра во внешний модуль ядра, но эффективная поддержка всех системных вызовов в этом случае затруднена.

Таким образом, выбранный в настоящей работе подход позволяет, с одной стороны, применять реализованный фильтр для большого количества вызовов, с другой стороны, потери производительности будут не очень существенны.

Настоящая работа, помимо описанных в ней результатов, открывает перспективы разработки многослойной системы фильтров для ядра операционной системы. Кроме того, выше была описана одна из компонент системы активного аудита, которая использует разработанные программные средства.

Литература

[1] Ахманов С. А. Фильтр системных вызовов для ОС Linux. Сборник студенческих работ по программе «СКИФ». 2003 г.

[2] Общие Критерии Оценки Безопасности Информационных Технологий. `http://www.commoncriteria.org`.

[3] Проект Overloader. `http://bdolez.free.fr/dev`.

[4] Проект System Call Tracker. `http://syscalltrack.sourceforge.net`.

[5] Робачевский А. М. Операционная система UNIX. БХВ — Санкт-Петербург. 1999.

[6] Ядро ОС Linux. `http://www.kernel.org`.

[7] Akhmanov S. A. System Calls Filter for OS Linux. ACS'2002.

[8] Pomerantz Ori. Linux Kernel Module Programming Guide. 1999.

[9] Mitchel Mark, Oldham Jeffrey, Samuel Alex. Advanced Linux Programming. New Riders Publishing, 2001.

[10] Pluggable Authentication Modules. `http://www.kernel.org/pub/linux/libs/pam`.

[11] Pluggable Non Interactive Authentication Modules. `http://www.msu.ru/pniam/pniam.html`.

[12] UnixBench — универсальные тесты производительности для Unix-систем. `http://www.tux.org/pub/tux/niemi/unixbench`.

Разработка механизмов контроля и распределения ресурсов в ОС Linux (на уровне пользователя)

Д. А. Надежкин, Д. А. Раевский

1 Введение

Один из основных критериев, которому должна удовлетворять современная операционная система (ОС) - надежность. Важным условием, без которого обеспечить такую работу ОС практически невозможно, является наличие механизмов контроля и распределения системных ресурсов (оперативной и дисковой памяти, процессорного времени, сетевого трафика). Во-первых, современные операционные системы являются многопользовательскими и многозадачными, что налагает определенные ограничения при использовании доступных ресурсов, так как их одновременно приходится делить между всеми выполняемыми на компьютере задачами. Во-вторых, наличие механизмов контроля ресурсов

позволит защитить систему от многих программных ошибок (как случайных, так и умышленных). В-третьих, успех различных сетевых атак (например, атак на отказ в обслуживании) во многом так же обусловлен отсутствием эффективной системы распределения ресурсов ОС. При этом реализация данных механизмов не должна сказываться на производительности компьютерной системы, а усиление надежности ОС за счет их введения не должно быть реализовано в ущерб удобству использования.

1.1 Основные требования

К числу основных требований, которым должна удовлетворять система распределения ресурсов, чтобы можно было избежать всех вышеперечисленных проблем, можно отнести следующие

- Необходимо строго разделять (и контролировать этот процесс) ресурсы операционной системы между всеми выполняющимися в системе задачами (на основе механизма ограничений - квот), так как одновременно могут работать несколько ресурсоемких процессов.
- Следует блокировать любые процессы, которые преднамеренно или ввиду ошибки могут привести к исчерпанию ресурсов.
- Процесс распределения должен легко настраиваться в соответствии с любыми требованиями, предъявляемыми системным администратором, что позволит сконфигурировать использование ресурсов в полном соответствии с ними. Необходимо, чтобы администратор мог настроить механизмы контроля (а точнее ограничения на использование ресурсов) для каждого пользователя в отдельности (в зависимости от приоритета данного пользователя).
- Часть ресурсов ОС должна резервироваться с тем, чтобы избежать ситуации, когда администратор не сможет восстановить функциональность системы, так как не хватает, например, свободной памяти.

1.2 Способы реализации

Так как в операционной системе Linux не только приложения, но и само ядро распространяются в виде исходных кодов, то при создании механизмов контроля и распределения ресурсов можно пойти двумя путями. а именно - реализовать данные возможности на уровне ядра ОС или на уровне приложений.

Первый способ представляется более корректным и обладает следующими преимуществами:

- достаточно внести изменения в код ядра, а так же в код всего нескольких приложений, которые являются точками входа (например, утилиты login, su и т.д.), в которых и будет происходить процесс установки ограничений, заданных администратором
- при реализации на уровне ядра обойти систему контроля будет гораздо сложнее, чем при ее реализации на уровне приложений
- работая на уровне ядра, возможно более эффективно управлять процессом создания/уничтожения новых задач, выделением памяти и т.п.
- уровень производительности будет выше, так как иначе, в процессе работы нужно было бы постоянно пользоваться системными вызовами, то есть переключаться в ядро и обратно, что неблагоприятно сказывается на скорости работы.

2 Существующие решения

Рассмотрим ядро Linux версии 2.4.22 (последнее на момент подготовки данной публикации ядро ветки 2.4). В нем реализованы очень мощные механизмы контроля ресурсов, называемые rlimits (от Resource limits [3]). Они позволяют задавать тип контролируемого ресурса (память, процессорное время, и т.д.), ограничения на него, а так же предоставляют несколько системных вызовов, которые устанавливают лимиты на ресурсы. Однако основной (и очень существенный недостаток) rlimit-ов в том, что процесс контроля и распределения ресурсов операционной системы идет только на уровне каждого отдельно взятого приложения.

Вместе с тем даже если каждый процесс не превысит ограничений, установленных для него администратором, в сумме все работающие задачи вполне могут привести к исчерпанию свободных ресурсов. С учетом этого обстоятельства хотелось бы задавать ограничения суммарно по всем задачам пользователя. Анализ работ на данном направлении показывает, что эффективных механизмов, позволяющих добиться этого, в ядре ОС Linux нет.

2.1 Resource limits

Как было написано выше, механизмы rlimit-ов очень удобны и функциональны, поэтому они были взяты за основу разработанной нами системы контроля и распределения ресурсов, которая была названа ulimits (от User limits). Перед тем, как перейти к описанию механизмов User limits, опишем систему Resource limits более подробно.

При создании каждого нового процесса для него устанавливаются ограничения на каждый тип ресурса, которые нельзя превышать. Данные ограничения задаются в виде следующей структуры, которая присутствует для каждой задачи:

```
struct rlimit {
    unsigned long rlim_cur;
    unsigned long rlim_max;
},
```

где `rlim_cur` и `rlim_max` - соответственно мягкое (soft) и жесткое (hard) ограничения на ресурс.

Мягкая граница - это текущий лимит на использование ресурса, а жесткая - это максимальный предел потребления ресурса. Каждый процесс может изменить значение мягкой границы, уменьшив его или увеличив вплоть до жесткой границы (но не более), которую можно только уменьшить. Увеличить hard-ограничение могут только процессы, обладающие правами суперпользователя.

Тип контролируемого ресурса задается константой. Наиболее интересные из них представлены ниже:

```
RLIMIT_NPROC /* количество процессов */
RLIMIT_CPU /* процессорное время */
RLIMIT_DATA /* размер области данных */
RLIMIT_STACK /* размер стека */
RLIMIT_NOFILE /* количество файловых дескрипторов */
```

Первоначально квоты устанавливаются при инициализации системы, а вновь создаваемые задачи наследуют их от процессов-родителей. Но в дальнейшем, данные ограничения могут быть изменены. Если нет необходимости в ограничении использования какого-то ресурса для данного процесса, то можно установить мягкую и жесткую границы для этого ресурса в значение `RLIM_INFINITY`, что означает неограниченное использование. Точнее, в данном случае уже физические ограничения системы (объем оперативной памяти, место на диске и т.п.) будут определять реальный предел.

Для получения информации о текущих квотах на ресурс или для изменения этих лимитов в ядре операционной системы Linux реализованы системные вызовы [5] `sys_getrlimit()` и `sys_setrlimit()`, описанные следующим образом:

- `int sys_getrlimit(int resource, struct rlimit *rlim)` — получение информации об установленных ограничениях на ресурс resource для текущего процесса;
- `int sys_setrlimit(int resource, struct rlimit *rlim)` — изменение мягкой и жесткой границы ресурса resource для текущего процесса.

Используя эти системные вызовы, процессы могут узнавать или изменять свои квоты на ресурсы.

3 User limits

Как было написано выше, нами были разработаны механизмы, позволяющие контролировать ресурсы суммарно по всем задачам каждого пользователя, которые мы назвали User Limits. В процессе реализации системы контроля ресурсов пользователя возникло несколько важных вопросов, которые необходимо решить, а именно:

- какие ресурсы необходимо контролировать;
- по какому принципу организовывать данный контроль;
- где и в каком виде хранить необходимую информацию;
- способы управления и настройки данной системы контроля.

Рассмотрим эти вопросы более подробно. Ресурсов, которые можно контролировать по всем задачам пользователя, довольно много, но пока был реализован контроль следующих из них: количество процессов пользователя, процессорное время, выделяемое суммарно всем задачам пользователя, а так же количество файловых дескрипторов, принадлежащих пользователю. Почему были выбраны именно эти ресурсы? Количество процессов нужно строго отслеживать, так как каждый новый процесс требует памяти, процессорного времени, файловых дескрипторов и т.п. Процессорное время необходимо контролировать, чтобы его можно было разделить между всеми пользователями по каким-то критериям. Файловые дескрипторы были выбраны потому, что, не смотря на название, они отвечают за работу с файлам, сокетами (sockets), каналами (pipes) и др.

При реализации системы контроля ресурсов, во-первых, необходимо определить новые типы данных, описать константы и функции. Во-вторых, для удобства работы с ulimit-ами было добавлено несколько дополнительных пунктов меню при конфигурировании ядра операционной системы, которые позволяют настроить систему контроля: включить (или выключить) поддержку user limits в ядре, выбрать способ работы механизма контроля (об этом ниже), а так же при необходимости ограничить максимальное число пользователей, которые могут работать в системе. В-третьих, реализованы новые системные вызовы, через которые осуществляется конфигурирование ulimit-тов.

По аналогии с rlimit-ми были описаны следующие константы, каждая из которых соответствует определенному типу ресурса, контролируемому на уровне пользователя:

```
ULIMIT_CPU /* процессорное время */
ULIMIT_NPROC /* количество процессов */
ULIMIT_NOFILE /* количество файловых дескрипторов */
```

Была введена следующая структура, описывающая квоты на ресурсы, установленные для пользователя:

```
struct ulimit {
    unsigned long ulim_cur;
    unsigned long ulim_max;
}.
```

Смысл полей этой структуры аналогичен полям `rlim_cur` и `rlim_max` структуры rlimit. Однако, в данном случае, это ограничение на все пользовательские процессы сразу. При создании первого процесса пользователя (когда он регистрируется в системе) создается структура `user_struct`, в которую система и прописывает ограничения (заданные администратором) на все типы ресурсов. В данной структуре храниться так же число, указывающее общий объем ресурсов, который суммарно используют все задачи пользователя.

Структура `user_struct` имеет следующий вид:

```
struct user_struct {
    atomic_t __count;
    struct ulimit cpu_lim;
    atomic_t processes;
```

```
        struct ulimit processes_lim;
        atomic_t files;
        struct ulimit files_lim;
        uid_t uid;
    },
```

где

- `__count` — количество процессов, ссылающихся на данную структуру;
- `cpu_lim` — ограничения на суммарное процессорное время для всех задач пользователя;
- `processes` — количество действующих процессов пользователя;
- `processes_lim` — ограничение на количество пользовательских процессов;
- `files` — количество используемых пользователем в данный момент файловых дескрипторов;
- `files_lim` — ограничение на количество файловых дескрипторов;
- `uid` — идентификатор пользователя.

При создании нового процесса проверяется, не превышены ли квоты на каждый тип ресурса, и, в случае положительного результата процесс создается. При выделении памяти, создании файла и других подобных операциях сначала проверяется, не будут ли превышены лимиты, и только тогда действие разрешается. Далее, были изменены функции `do_fork()` (отвечает за создание нового процесса), `sys_setuid()` (изменение владельца процесса), `exec_usermodehelper()` и др. Для контроля количества файловых дескрипторов необходимо модифицировать функции работы с файлами, сокетами, каналами и т.п. Подвергся модификации и код системных вызовов `sys_setrlimit()` и `sys_getrlimit()`, так как механизм rlimit-ов должен плотно взаимодействовать с новым механизмом ulimit-ов. Конечно же, необходимо добавить в ядро новые системные вызовы, при помощи которых можно будет изменять квоты на ресурс, а так же получать их текущее значение. Были добавлены два новых системных вызова `sys_setulimit()` и `sys_getulimit()`, которые работаю по аналогии с такими же вызовами для `resource limits`:

- `int sys_getulimit(int resource, struct ulimit *ulim)` - получение информации об установленных квотах на ресурс resource для текущего пользователя (для всех его задач);
- `int sys_setulimit(int resource, struct ulimit *ulim)` - изменение мягкой и жесткой границы ресурса resource для текущего пользователя (для всех его задач).

В процессе реализации системы контроля возникли и другие трудности, которые, на первый взгляд, не заметны. Например, неясность, которая возникает в случае, если ядро получает запрос на смену владельца текущего процесса? Задача ядра - проверить, будет ли превышен лимит на количество процессов (или на процессорное время) у нового пользователя при смене владельца процесса. Но процессу принадлежат открытые файловые дескрипторы, которые, соответственно, тоже меняют владельца. И это необходимо учитывать. Не следует забывать, что при создании нового процесса он получает копии дескрипторов, принадлежащих родителю. И если происходит изменение владельца процесса-родителя или процесса-потомка, то может возникнуть путаница с используемыми процессом файловыми дескрипторами, то есть с файлами, сокетами, каналами.

4 Заключение

Представленная в настоящей публикации реализация системы контроля и распределения ресурсов имеет ряд преимуществ. Так, несмотря на то, что количество контролируемых типов ресурсов в представленной версии системы невелико (количество процессов и файловых дескрипторов, процессорное время), общая структура разработанных механизмов контроля позволяет по необходимости добавлять поддержку других ресурсов. Используя два новых системных вызова, можно создавать приложения, взаимодействующие с механизмами контроля ресурсов. Несомненным достоинством использованного

в работе подхода является реализация системы контроля на уровне ядра, что обеспечивает минимальное падение производительности. Добавленные системные вызовы упрощают работу с механизмами контроля. Возможность конфигурирования системы для отдельно каждого пользователя позволяет добиться большой гибкости при конфигурировании.

Литература

[1] Understanding the Linux Kernel. Daniel P. Bovet, Marco Cesati. O'Reilly, 2000.

[2] Linux Kernel 2.4 Internals. Tigran Aivazian. 2002.

[3] Робачевский А. М. Операционная система UNIX. БХВ — Санкт-Петербург. 1999.

[4] Ядро ОС Linux. `http://www.kernel.org`

[5] Operating System Concepts. Silberschatz, Galvin. 2002.

[6] The Linux kernel. Andries Brouwer. 2003.

Метод анализа динамики развития атаки

С. С. Корт, Е. А. Рудина

Для начала дадим несколько определений. *Атака* на компьютерную систему – это действие, предпринимаемое злоумышленником, которое заключается в поиске и использовании той или иной уязвимости. Под *вторжением* будем понимать нарушение безопасности, состоящее из одной или нескольких атак. Типовой сценарий вторжения состоит из следующих этапов:

1. *Этап сбора информации.* На этапе сбора информации нарушителя может интересовать информация об атакуемой системе, в том числе:

- топология сети, в которой функционирует атакуемая система;
- тип ОС на атакуемых хостах;
- функционирующие на хостах сервисы;
- дополнительная информация об атакуемых хостах.

2. *Этап непосредственной атаки.* На основании информации об атакуемом объекте, собранной в результате выполнения предыдущего этапа, нарушитель может начать атаку на систему. На данном этапе используются типовые уязвимости в системных сервисах или ошибки в администрировании системы. Успешным результатом использования уязвимостей обычно является получение нарушителем прав на атакованном хосте, получение файла паролей, отказ в обслуживании атакуемого хоста и т. д.

3. *Этап консолидации.* После того как нарушитель осуществил с использованием атаки на систему проникновение в нее, обычно начинается использование скомпрометированного хоста - этап консолидации. Данная стадия вторжения может быть подразделена на две логические фазы: консолидация и распространение вторжения. Кроме того, к данной фазе можно отнести и действия нарушителя, связанные с реализацией угроз безопасности (например, доступ к защищаемой информации).

Существующие методы обнаружения вторжений можно в общем случае охарактеризовать следующим образом:

- сигнатурные методы выявления атак направлены на выявление составных элементов вторжения и не пытаются объединить полученные результаты в единую картину вторжения;

- существующие методы искусственного интеллекта, дополняющие сигнатурные методы выявления атак, направлены на выявление сложных атак и также не учитывают этапы вторжения;
- методы выявления аномалий исследуют отклонения в поведении пользователей и не позволяют отследить картину вторжения.

В СЦЗИ СПбГПУ была разработана система анализа вторжений, принципы работы которой рассмотрены в данной статье. Система при своей работе использует два алгоритма:

- Алгоритм, исследующий динамику развития атаки в соответствии с типовыми сценариями вторжения и использующий для этого алгоритм, основанный на автомате состояний.
- Алгоритм, основанный на поиске аномалий.

В основу алгоритма анализа динамики развития атаки в соответствии с типовыми сценариями вторжения заложены типовые сценарии атак, основанные на взаимосвязи ее различных этапов. Этот алгоритм анализирует состояние внешних хостов по отношению к защищаемому. При этом внешний хост может быть охарактеризован следующим образом:

1. Неизвестный хост – хост, обращавшийся к защищаемому хосту в течение краткого периода времени, и не производивший атакующих действий;
2. Доверенный хост - хост, обращавшийся к защищаемому хосту в течение длительного периода времени, и не производивший атакующих действий;
3. Подозрительный хост – хост, сканировавший или атаковавший защищаемый хост.

Данные для анализа с использованием данного метода, управляющие переходами автомата, поступают от системы обнаружения атак Snort. Метод описывает отношения «атакующий хост» - «защищаемый хост» в виде автомата конечных состояний (показан на рисунке 1). Переходы данного автомата можно описать выражением:
Transition (Action, ServiceName, StandartRecone, OSDepended, UserDepended, FSDepended, UsedTool)
Таким образом, в описании перехода автомата определены следующие переменные:

1. Action – описывает действие нарушителя; возможны следующие типы действий:

a) сканирование – сервиса (ReconService), операционной системы (ReconOS), информации о файловой системе хоста (ReconFS), учетных записях пользователей (ReconUser); одно и то же сканирование может одновременно относится к нескольким типам;

b) атаки отказа в обслуживании – сервиса (DOSService), хоста (DosHost); атаки на учетную запись – администратора (ServiceAdminAttack), пользователя (ServiceUserAttack), атаки чтения объекта файловой системы (ReadFile), записи в объект файловой системы (WriteFile); каждая атака может быть только одного из перечисленных типов;

c) консолидации – с использованием известного троянского ПО(ConsolidationTroyan), выполнение команд на защищаемом сервере (Consolidation).

2. ServiceName – имя сервиса используемого нарушителем для атаки; в том случае, если действие нарушителя не зависит от конкретного сервиса, используется ключевое слово «MISC».

3. StandartRecone – использование стандартных методов обращения к защищаемому хосту; может принимать значения «Yes», «No».

4. OSDepended – зависимость фазы атаки от знаний об операционной системе, установленной на защищаемом хосте; может принимать значения «Yes», «No».

5. UsedTool – использование известных средств нападения может принимать значения «Yes», «No».

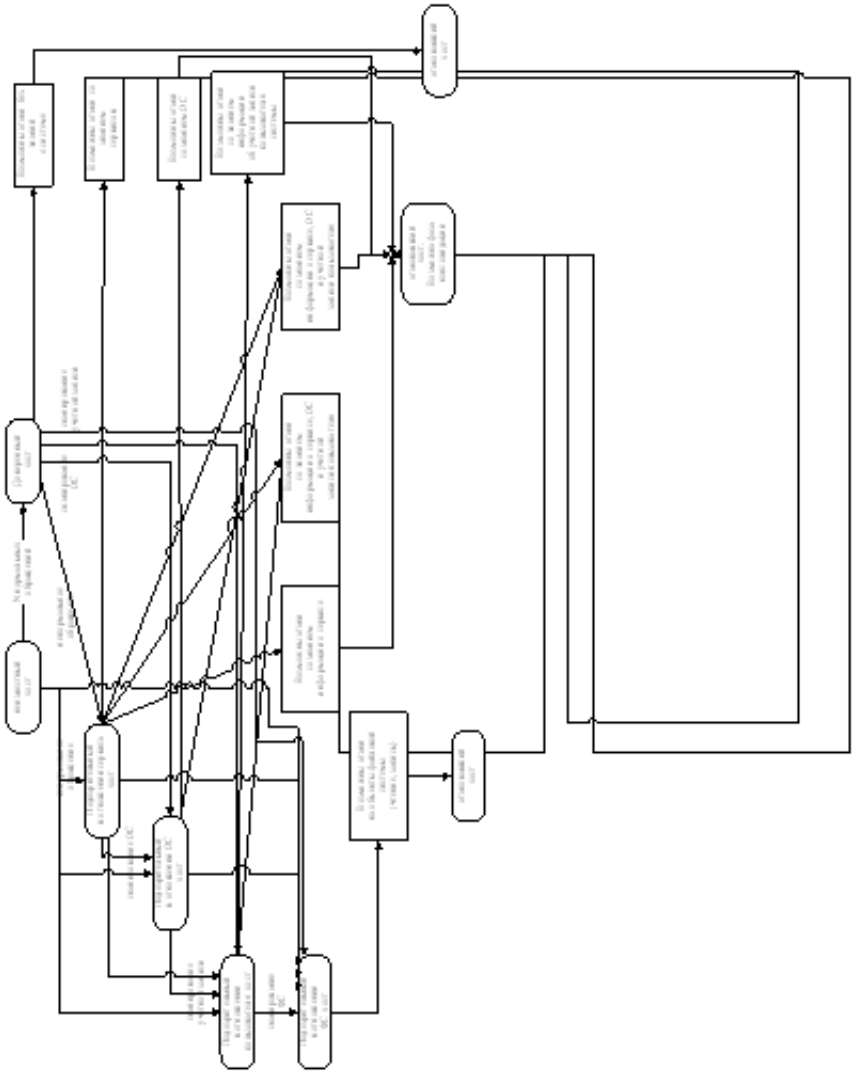


Рис. 1:

Для каждого сообщения, классифицированного как сообщение об атаке или фазе консолидации, производится верификация сценария в соответствии с текущим состоянием обращающегося хоста по отношению к сервису, ОС, подсистеме учетных записей и файловой системе защищаемого хоста. Верификация означает сопоставление атрибутов «подозрительности» обращающегося хоста с атрибутами, описывающими классифицированное действие. Признаком успешной (верифицированной) атаки является обнаружение сигнатуры, соответствующей этапу непосредственной атаки, после обнаружения сигнатуры сканирования с нужными атрибутами. Признаком успешного (верифицированного) вторжения (конечное состояние автомата) является обнаружение сигнатуры, соответствующей этапу консолидации после обнаружения сигнатуры непосредственной атаки нужного типа.

Недостатком данного алгоритма является его зависимость от сигнатур, генерируемых системой обнаружения атак Snort. Атаки, не обнаруженные Snort, не будут оценены как часть вторжения. С целью получения более точной оценки вторжения метод анализа динамики развития атаки был дополнен вторым алгоритмом, основанным на выявлении аномалий.

Основой построения алгоритма выявления аномалий являются данные, образующие профиль поведения внешнего хоста по отношению к каждому сервису защищаемого хоста. В нашем случае профиль поведения составляли следующие данные:

a) тип запрашиваемых с помощью сервиса ресурсов защищаемого сервера; тип ресурса определяется на основании его расширения;

b) стандартное время обращений к сервису (сессия);

c) количество обращений к сервису за сессию;

d) количество ошибок при обращении к защищаемому сервису за сессию.

Описание соответствующих характеристик приведено в таблице 1.

Характеристика, используемая при аномальном анализе	Описание характеристики
Стандартное время обращения к сервису сервера	Интервалы времени
Запрашиваемые с помощью сервиса ресурсы	Множество типов запрашиваемых ресурсов
Количество обращений к сервису за сессию	Среднее количество обращений за сессию и среднеквадратическое отклонение
Количество ошибок при обращении к сервису за сессию	Среднее количество ошибок за сессию и среднеквадратическое отклонение

Таблица 1:

Для каждой из описанных характеристик статистика нормального поведения вычисляется при корректных обращениях к защищаемому хосту до того момента, когда хост переносится в список доверенных хостов. Текущая активность хоста оценивается в течение сессии. В качестве сессии рассматривается некоторый промежуток времени (один день). В методе предлагается переводить хост в список доверенных после N дней работы с защищаемым хостом с использованием любого сервиса (рекомендуется устанавливать параметр N равным не менее 30). После того, как хост был переведен в список доверенных хостов, начинается сравнение текущей активности хоста с составленным профилем нормального поведения хоста. Сравнение в данном случае выполняется по следующим критериям:

a) для множеств номеров портов и множества типов ресурсов – присутствие соответствующего элемента в данных о текущей активности в профиле нормального поведения;

b) для количества ошибок при обращении к сервису и количества обращений к сервису – попадание соответствующих характеристик текущей сессии, в интервал, рассчитанный в профиле нормального поведения;

c) для времени сессии – попадание текущей сессии в интервал, определенный в профиле.

В результате сравнения текущей активности с профилем нормального поведения выявляются отклонения в поведении внешнего хоста – аномалии. Кроме того, что алгоритм выявления аномалий выявляет отклонения в поведении внешнего хоста – аномалии, он дополняет алгоритм, исследующий динамику развития атаки в соответствии с типовыми сценариями вторжения, при оценке успеха вторжения следующим образом. Если хост инициировал атаку, обнаруженную с использованием метода анализа динамики атаки, после которой была зафиксировано аномальное поведение, вторжение считается успешным. После этого профиль нормального поведения считается недействительным. Кроме того, если от хоста была обнаружена сигнатура консолидации и было выявлено аномальное поведение, можно сигнализировать о возможной атаке, сигнатурой которой мы не обладаем, также оценивая вторжение как успешное. Если в поведении доверенного хоста после получения единичного сообщения, соответствующего сигнатуре сканирования или непосредственной атаки, не было отклонений от профиля или соответствующего сообщения о консолидации, то сообщение считается ложным.

Соответствующие условия оценки успеха вторжения приведены в таблице 2.

	Вторжение не считается успешным	Вторжение считается успешным
Сигнатура атаки	+	
Сигнатура консолидации	+	
Аномальное поведение	+	
Сигнатура атаки и сигнатура консолидации		+
Сигнатура атаки и аномальное поведение		+
Сигнатура консолидации и аномальное поведение		+

Таблица 2:

Использование данного подхода позволяет уменьшить количество ложных срабатываний (приводящих к необходимости создания нового профиля нормального поведения), связанных с аномальной активностью доверенного хоста, не ведущей явно к атаке на защищаемый хост, а также с единичными сигнатурами атак.

Таким образом, разработанный метод анализа динамики развития атаки позволяет:

a) определить атаки возможные на сервис в соответствии со сценарием (предсказание сценария атаки);

b) оценить сценарий развития атаки (верификация сценария атаки).

Доступ к базам данных без раскрытия запроса

Э. Э. Гасанов, Г. А. Майлыбаева

1 Понятие PIR (Private Information Retrieval) протокола

Рассмотрим протокол с $k+1$ участником: пользователем и k серверами ($k \geqslant 1$), каждый из которых хранит один и тот же булев вектор $x = (x_1, \ldots, x_n)$ длины n — базу данных. Пользователь желает узнать значение i-го бита x_i этого вектора так, чтобы номер бита i не стал известен ни одному из серверов. При этом пользователь имеет возможность получать случайные булевы векторы, т.е. имеет некоторый генератор случайных последовательностей. Понятие PIR-протокола впервые было введено в [1]. Здесь мы приводим другое, строго формальное, определение PIR-протокола.

Для любого натурального n обозначим $[n] = \{1, \ldots, n\}$.

Протоколом доступа называется тройка $I = \langle Q, A, R\rangle$, где Q, A, R некоторые отображения $Q : [k]\times[n]\times\{0,1\}^s \to \{0,1\}^m$, $A : [k]\times\{0,1\}^n\times\{0,1\}^m \to \{0,1\}^p$, $R : [n]\times\{0,1\}^s\times\{0,1\}^{km}\times\{0,1\}^{kp} \to \{0,1\}$,

такие, что выполнено условие корректности: $\forall\, r \in \{0,1\}^s$, $\forall\, i \in [n]$

$$R(i, r, Q(1, i, r), \ldots, Q(k, i, r), A(1, x, Q(1, i, r)), \ldots, A(k, x, Q(k, i, r))) = x_i.$$

Содержательно протокол $I = \langle Q, A, R\rangle$ состоит из следующих шагов:

- Пользователь U, имея запрос i, вырабатывает случайную строку $r = (r_1, \ldots, r_s)$, для каждого $j \in [k]$ вычисляет $q^j = (q_1^j, \ldots, q_m^j) = Q(j, i, r)$ и посылает вектор q^j j-му серверу S_j.
- Каждый сервер S_j вычисляет $a^j = (a_1^j, \ldots, a_p^j) = A(j, x, q^j)$ и посылает вектор a^j пользователю.
- U вычисляет $x_i = R(i, r, q^1, \ldots, q^k, a^1, \ldots, a^k)$.

Величина $C(I) = k(m + p)$ называется *сложностью передачи протокола* $I = \langle Q, A, R\rangle$.

Протокол доступа $I = \langle Q, A, R\rangle$ называется *PIR-протоколом*, если выполняется условие защищенности: $\forall\, q \in \mathcal{Q}$, $\forall\, t \in [k]$, $\forall\, i, j \in [n]$

$$\mathsf{P}(Q(t, i, r) = q) = \mathsf{P}(Q(t, j, r) = q),$$

где $\mathcal{Q} = \{(q_1^j, \ldots, q_m^j) = Q(j, i, r) : j \in [k], i \in [n], r \in \{0,1\}^s\}$, P — равномерная вероятностная мера, заданная на $\{0,1\}^s$.

Условие корректности гарантирует, что пользователь получит нужный бит базы данных, а условие защищенности — что ни один из серверов по вектору q, который он получил, не сможет понять какой бит интересует пользователя.

Существование PIR-протоколов доказывается следующим примером простейшего PIR-протокола, у которого $Q \equiv 0$, т. е. $s = 0$ и $m = 0$, $A(j, x) = (x_{(j-1)\frac{n}{k}+1}, \ldots, x_{j\frac{n}{k}})$, $R(i, x) = x_i$. Сложность этого протокола равна n, а содержательно он состоит в том, что каждый сервер выдает пользователю свою часть базы данных, а пользователь собрав всю базу данных извлекает нужный бит.

Основной целью исследований в этой области является нахождение для заданных n и k PIR-протокола с минимальной сложностью передачи.

2 Лучшие результаты

- [1, B. Chor, O. Goldreich, E. Kushilevitz, M. Sudan, 1995 г.]. Предложен PIR-протокол для $O(\log n)$ серверов со сложностью передачи $O(\log^2 n \log\log n)$.
- [2, A. Ambainis, 1997 г.] Был получен PIR-протокол для k серверов, $k \geqslant 2$, со сложностью передачи $O(n^{1/2k-1})$.
- [3, A. Beimel, Y. Ishai, E. Kushilevitz, J. F. Raymond, 2002 г.] Разработан PIR-протокол для к серверов со сложностью передачи $O\left(n^{\frac{c \log\log k}{k \log k}}\right)$ для некоторой константы c.

3 Понятие частичного раскрытия

Один из способов сокращения сложности передачи — это допустить, что серверы могут частично раскрыть запрос, т. е. могут с достаточно большой вероятностью сделать предположение об области принадлежности запроса.

Степенью раскрытия протокола доступа $I = \langle Q, A, R\rangle$ назовем величину

$$D(I) = \max_{q \in \mathcal{Q}} \max_{j \in [k]} \max_{M \subseteq [n]} \frac{\sum_{i \in M} \mathsf{P}(Q(j, i, r) = q)}{\mid M \mid}.$$

Понятно, что для PIR-протокола степень раскрытия равна $1/n$.

Протокол доступа I называется *протоколом с частичным раскрытием*, если $D(I) > 1/n$.

Если $D(I) = 1$, то значит некоторый сервер может с вероятностью 1 определить значение запроса.

Обозначим через $\mathcal{I}(n,k)$ множество PIR-протоколов, а через $\mathcal{I}(n,k,d)$ множество протоколов доступа со степенью раскрытия не более, чем d, с k серверами и базой данных размером n бит.Обозначим

$$C(n,k) = \min_{I\in\mathcal{I}(n,k)} C(I), \quad C(n,k,d) = \min_{I\in\mathcal{I}(n,k,d)} C(I).$$

Следующая теорема отражает связь между PIR-протоколами и протоколами с частичным раскрытием.

Теорема 1. *Для любых натуральных k, n и любого вещественного числа $d \in [1/n, 1]$ выполнено $C(n,k,d) \leqslant C(]1/d[,k) + k]\log_2 dn[$, где $]a[$ — наименьшее целое не меньшее, чем вещественное a.*

Литература

[1] Chor B., Goldreich O., Kushilevitz E., Sudan M. Private information retrieval. In Proc. of 36th FOCS, 1995.

[2] Ambainis A. Upper bound on the communication complexity of private information retrieval. In Proc. of 24th ICALP, 1997.

[3] Beimel A., Ishai Y., Kushilevitz E., Raymond J. F. Breaking the $O(n^{1/2k-1})$ barrier for inforamtiontheoretic private information retrieval. 2002.

Избранные вопросы теории автоматов и их приложения в криптографии

А. В. Бабаш

В докладе раскрывается содержание следующего спецкурса автора.

Место теории моделируемости, тестирования и периодичности автоматов среди других дисциплин. Приложения теории автоматов к криптографическому анализу. Краткая история развития теории шифрующих автоматов. Литература по курсу.

Раздел 1. Моделирование конечных автоматов.

Тема 1. Классификация моделей конечных автоматов.

1. Приближенные модели автоматов, построенные на основе расстояния Хэмминга между их выходными последовательностями.
2. Модели автоматов, построенные на основе обработки их входных и выходных последовательностей с помощью функций.
3. Модели автоматов, построенные на основе обработки их входных и выходных последовательностей с помощью инициальных автоматов.
4. Модели автоматов, построенные на основе расстояния Хэмминга между их табличными заданиями.
5. Модели автоматов, построенные на основе обобщения понятия гомоморфизма автоматов.

Раздел 2. Периодичность конечных автоматов.

Тема 2 . Периодичность выходных последовательностей конечных автоматов.

1. Гарантирование периодов выходных последовательностей некоторых классов автоматов.

2. Гарантирование локальных периодов выходных последовательностей некоторых классов автоматов.

3. Гарантирование мер приближенной периодичности выходных последовательностей автоматов некоторых классов.

4. Приближенная σ-периодичность и изопериодичность функционирования некоторых классов автоматов (σ - бинарное отношение).

Раздел 3. Инварианты автоматов.

Тема 3. Закрытые эксперименты с автоматами. Инварианты автоматов.

1. Закрытые эксперименты с автоматами по распознавания информации о входном слове автомата и начальном состоянии.

2. Случайное тестирование конечного автомата по входной и выходной последовательностям.

Литература

[1] Бабаш А. В. Приближенные модели перестановочных автоматов. РАН, Дискретная математика, т. 9, вып. 1. М.: ТВП, 1997, стр. 103–122.

[2] Бабаш А. В. Решение автоматных уравнений с искажениями в функции переходов автомата. Обозрение прикладной и промышленной математики. Пятая Всероссийская школа-коллоквиум по стохастическим методам. Т. 5, вып. 2, М.: ТВП, 1998, стр. 198–199.

[3] Бабаш А. В. Решение автоматных уравнений с искажениями в функции переходов автомата. М., 2002, принята к опубликованию в журнал «Проблемы передачи информации».

[4] Бабаш А. В. О некоторых инвариантах конечного автомата. Обозрение прикладной и промышленной математики. / Пятая Международная Петрозаводская конференция. Тезисы докладов. Т. 7, вып. 1, М.: ТВП, 2000, стр. 86–87.

[5] Бабаш А. В. G-изопериод выходной последовательности автономного последовательного соединения автоматов. Пятая Международная Петрозаводская конференция. Тезисы докладов. Т. 7, вып. 1, М.: ТВП, 2000, стр. 87–88.

[6] Бабаш А. В. Локальные периоды выходных последовательностей некоторых классов автоматов. / Обозрение прикладной и промышленной математики. Пятая Международная Петрозаводская конференция. Тезисы докладов. Т. 7, вып. 1, М.: ТВП, 2000, стр. 88–89.

[7] Бабаш А. В. Неотличимость состояний конечных автоматов относительно инициальных автоматов. / Обозрение прикладной и промышленной математики. Первый Всероссийский симпозиум по прикладной и промышленной математике. Тезисы докладов. Т. 7, вып. 2, М.: ТВП, 2000, стр. 306–307.

[8] Бабаш А. В. Частичные гомоморфизмы автоматов. / Обозрение прикладной и промышленной математики. Тезисы докладов. Т. 7, вып. 2, М.: ТВП, 2000, стр. 307–309.

[9] Бабаш А. В. Многозначные гомоморфизмы конечных автоматов. / Обозрение прикладной и промышленной математики. Четвертая Всероссийская школа-коллоквиум по стохастическим методам. Тезисы докладов. Т. 4, вып. 3, М.: ТВП, 1997, стр. 321–322.

[10] Бабаш А. В. Изопериоды выходных последовательностей автономных автоматов. Методы и технические средства обеспечения безопасности информации. Санкт-Петербург, 2000, стр. 88–90.

[11] Бабаш А. В. Приближенные периоды выходных последовательностей одного класса автономных автоматов. Санкт-Петербург, 2000, стр. 91–93.

[12] Бабаш А. В. О периодичности последовательности состояний автомата, отвечающей его начальному состоянию и входной периодической последовательности. М., 2001, Принята к опубликованию в журнале «Дискретная математика».

[13] Бабаш А. В. Локальное восстановление входных слов автоматов по начальным и заключительным состояниям. / Обозрение прикладной и промышленной математики. Второй Всероссийский симпозиум по прикладной и промышленной математике. Тезисы докладов. Т. 8, вып. 1, М.: ТВП, 2001, стр. 93–94.

[14] Бабаш А. В. Неотличимость состояний конечного автомата относительно функции, заданной на его входных и выходных словах. / Обозрение прикладной и промышленной математики. Второй Всероссийский симпозиум по прикладной и промышленной математике. Тезисы докладов. Т. 8, вып. 1, М.: ТВП, 2001, стр. 94–95.

[15] Бабаш А. В. Слабая автономность конечных автоматов. / Обозрение прикладной и промышленной математики. Второй Всероссийский симпозиум по прикладной и промышленной математике. Тезисы докладов. Т. 8, вып. 1, М.: ТВП, 2001, стр. 95–96.

[16] Бабаш А. В. Случайное тестирование конечного автомата по входной и выходной последовательностям. / Обозрение прикладной и промышленной математики. Тесисы докладов. 2002, передана для опубликования.

[17] Бабаш А. В. О восстановлении информации о входном слове перестановочного автомата Медведева по начальным и заключительным состояниям. М., 2002, передана для опубликования в журнал «Проблемы передачи информации».

[18] Бабаш А. В. Частичные изоморфизмы конечных автоматов. / Обозрение прикладной и промышленной математики. Второй Всероссийский симпозиум по прикладной и промышленной математике. Тезисы докладов. Т. 8, вып. 1, М.: ТВП, 2001, стр. 95–96.

[19] Бабаш А. В., Шанкин Г. П. Криптография. М., СОЛОН-Р, 2002.

[20] Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. М.: Наука, 1985.

[21] Балакин Г. В. Введение в теорию случайных систем уравнений. Труды по дискретной математике. 1997, ТВП, т. 1, стр. 1–18.

Геометрический подход к построению запретов k-значных функций

Н. В. Никонов

В действительной области пересечение множества решений системы неравенств

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n \geqslant b_1 \\ \ldots \\ a_{k1}x_1 + \ldots + a_{kn}x_n \geqslant b_k \end{cases} \tag{1}$$

с n-мерным единичным кубом задается системой (2)

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n \geqslant b_1 \\ \ldots \\ a_{k1}x_1 + \ldots + a_{kn}x_n \geqslant b_k \\ 0 \leqslant x_1 \leqslant 1 \\ \ldots \\ 0 \leqslant x_n \leqslant 1 \end{cases} \tag{2}$$

В частности, система (2) может содержать 0-1 решения и задавать, тем самым, некоторую булеву функцию, равную 1 на этих вершинах-решениях. Такое представление булевых функций для ряда прикладных задач дает определенные аналитические и алгоритмические преимущества (см. [1]).

Для перехода от булевых функций к k-значным, рассмотрим действие на систему (2) линейного преобразования растяжения $y = (k-1)x$. Получим новую систему:

$$\begin{cases} a_{11}y_1 + \ldots + a_{1n}y_n \geqslant b_1(k-1) \\ \ldots \\ a_{k1}y_1 + \ldots + a_{kn}y_n \geqslant b_k(k-1) \\ 0 \leqslant y_1 \leqslant k-1 \\ \ldots \\ 0 \leqslant y_n \leqslant k-1 \end{cases} \tag{3}$$

Очевидно, что система (3) совместна тогда и только тогда, когда совместна система (2).

Предложенная операция растяжения легла в основу построения классов k-значных функций с запретами исходя из булевых функций, имеющих запрет. Построение базируется на нахождении для булевой функции запретной комбинации, формировании на ее основе несовместной системы (2) и переходе к k-значному случаю преобразованием системы (2) в систему (3) с помощью операции растяжения.

Понятие запрета представляет интерес для различных прикладных задач, связанных с анализом дискретных узлов переработки информации и изучалось во многих работах. Напомним, что булева (или k-значная) функция $f(x_1, \ldots, x_n)$ имеет запрет $\gamma_1, \gamma_2, \ldots, \gamma_N$, если порожденная этой функцией система уравнений вида (4)

$$\begin{cases} f(x_1, \ldots, x_n) = \gamma_1 \\ f(x_2, \ldots, x_{n+1}) = \gamma_2 \\ \ldots \\ f(x_N, \ldots, x_{n+N-1}) = \gamma_N \end{cases} \tag{4}$$

- несовместна (см. [2]). Для доказательства несовместности системы (4) в булевом случае предлагается каждое уравнение системы заменить на систему неравенств (1), содержащую все решения данного уравнения и построить для всех таких уравнений результирующую систему линейных неравенств. Используя математический аппарат действительных соотношений можно для определенных классов таких систем установить их несовместность, тем самым строго доказать, что комбинация знаков $\gamma_1, \gamma_2, \ldots, \gamma_N$ - запрет. Далее, действие операции растяжения трансформирует данное доказательство из булевой области в k-значную, описывая, тем самым, класс функций k-значной логики, имеющих запрет. Таким образом, каждая булева функция, имеющая запрет, приводит к построению k-значных функций с запретами для различных значений k в зависимости от параметра k в операции растяжения.

В докладе приводятся примеры построения классов k-значных функций с запретами на базе булевых функций. Показывается, что одна булева функция может быть источником генерации различных классов k-значных функций с запретом.

Литература

[1] Балакин Г. В., Никонов В. Г. Методы сведения булевых уравнений к системам пороговых соотношений. Обозрение прикладной и промышленной математики, сер. «Дискретная математика», 1994, т. 1, в. 3, с. 389–401.

[2] Сумароков С. Н. Запреты двоичных функций и обратимость для одного класса кодирующих устройств. Обозрение прикладной и промышленной математики, сер. «Дискретная математика», 1994, т. 1, в. 1, с. 33–55.

О классификации всех булевых функций 3-х переменных с запретами и их связи с классами

k-значных функций, имеющих запрет

Н. В. Никонов

Предложенный в предыдущем докладе подход, позволяющий на базе любой булевой функции с запретом (см. [1]) строить классы k-значных функций с запретами, делает актуальным проведение классификационных исследований булевых функций, имеющих запреты. При проведении таких классификационных исследований необходимо для исследуемых булевых функций найти доказательство наличия запрета, основанное на построении для выделенной запретной комбинации несовместной системы линейных неравенств. Каждая такая несовместная система приведет к построению бесконечного класса k-значных функций с запретом для любого $k \geqslant 3$.

В геометрическом смысле исходной системе неравенств, задающей булеву функцию, соответствует система разделяющих плоскостей, вырезающих в n-мерном единичном кубе полиэдр, содержащий все единичные (или нулевые) вершины функции. Таких полиэдров, в равной мере позволяющих доказать наличие запрета, можно построить множество, поэтому при проведении классификационных исследований разделяющие плоскости задаются семействами, с включением изменяющихся параметров, что приводит к построению различных ветвей в генерации k-значных функций.

Автором проведены полные классификационные исследования всех булевых функций 3-х переменных с запретами (всего 240 функций без учета принадлежности к классам эквивалентности по наличию запрета) и составлен соответствующий каталог на базе принципов, предложенных в работе [2]. Для каждой булевой функции с помощью систем линейных неравенств найдено доказательство наличия обобщенного запрета (учитывающего произвольные разности расстояний между существенными переменными функции), а также запрета при равных расстояниях между существенными переменными. Все такие системы линейных неравенств задаются параметрически с указанием пределов изменения параметров и приводят к синтезу k-значных функций с запретом для любого $k \geqslant 3$.

В составленном каталоге для каждой булевой функции выделен самостоятельный раздел. Приводится геометрическое представление булевой функции, ее задание в виде систем линейных неравенств с изменяющимися параметрами. Для запретной комбинации дается строгое доказательство несовместности соответствующей системы линейных неравенств. В каждом случае из доказательства несовместности систем неравенств определяются пределы изменения параметров в задании булевой функции, обеспечивающие наличие запрета. Указанные пределы изменения параметров приводят к генерации классов k-значных функций с запретами.

В качестве примера рассмотрим раздел каталога, соответствующий булевой функции

$$f(x_1, x_2, x_3) = x_1x_2 \vee x_1x_3 \tag{1}$$

(см. рис. 1).

Приведем задание функции (1) с помощью систем линейных неравенств:

$$f = 1 \iff \begin{cases} x_1 \geqslant \delta, & 0 < \delta \leqslant 1 \\ x_2 + x_3 \geqslant \varepsilon_1, & 0 < \varepsilon_1 \leqslant 1 \end{cases}$$

Докажем, что комбинация выходных знаков

$$1\overbrace{\ldots}^{l_1}0\underbrace{\ldots}_{l_2}0\overbrace{\ldots}^{l_1}1 \quad (l_1, l_2 \text{ — любые}) \tag{2}$$

- запрет данной функции.

Рассмотрим комбинацию выходных знаков (2). В обозначениях, введенных на диаграмме

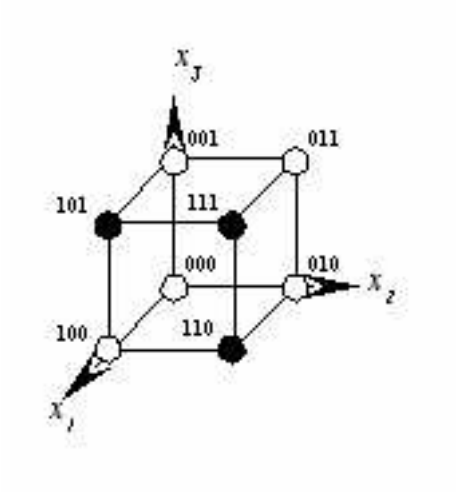

x_3
001
011
101
111
000
010
x_2
100
110
x_1

Рис. 1:

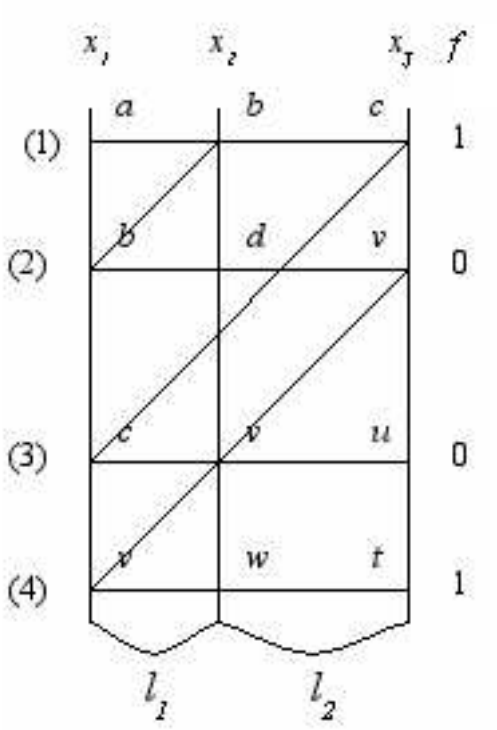

x_1
x_2
x_3
f
a
b
c
(1)
1
b
d
v
(2)
0
c
v
u
(3)
0
v
w
t
(4)
1
l_1
l_2

Рис. 2:

(см. рис. 2), последовательность знаков (2) порождает систему неравенств:

$$\left\{\begin{array}{l} 1\left\{\begin{array}{l} a \geqslant \delta \\ b+c \geqslant \varepsilon_1 \end{array}\right. \\ 0\left\{\begin{array}{l} -b-v \geqslant \varepsilon_2 - 2 \\ -b-d \geqslant \varepsilon_3 - 2 \end{array}\right. \\ 0\left\{\begin{array}{l} -c-u \geqslant \varepsilon_2 - 2 \\ -c-v \geqslant \varepsilon_3 - 2 \end{array}\right. \\ 1\left\{\begin{array}{l} v \geqslant \delta \\ w+t \geqslant \varepsilon_1 \end{array}\right. \end{array}\right. \Longrightarrow \left\{\begin{array}{l} b+c \geqslant \varepsilon_1 \\ -b-v \geqslant \varepsilon_2 - 2 \\ -c-v \geqslant \varepsilon_3 - 2 \\ 2v \geqslant 2\delta \end{array}\right. .$$

Складывая неравенства полученной системы, имеем: $0 \geqslant 2\delta + \varepsilon_1 + \varepsilon_2 + \varepsilon_3 - 4$.

Получаем противоречие при выполнении неравенства $2\delta + \varepsilon_1 + \varepsilon_2 + \varepsilon_3 - 4 > 0$.

Итак, комбинация знаков выходной последовательности (2) - запрет исходной функции (1) при выполнении условия $2\delta + \varepsilon_1 + \varepsilon_2 + \varepsilon_3 - 4 > 0$.

Литература

[1] Сумароков С. Н. Запреты двоичных функций и обратимость для одного класса кодирующих устройств. Обозрение прикладной и промышленной математики, сер. «Дискретная математика», 1994, т. 1, в. 1, с. 33–55.

[2] Никонов В. Г. Пороговые представления булевых функций. Обозрение прикладной и промышленной математики, сер. «Дискретная математика», 1994, т. 1, в. 3, с. 402–457.

1. Латинские квадраты широко используются в теории кодирования, планирования эксперимента, связи в секретных системах [4, 5]. При конструктивном задании латинского квадрата широко используется аналитический способ задания его с помощью функций, определяющих по номеру строки и номеру столбца значение соответствующего элемента квадрата. При этом не требуется запоминание (хранение) латинского квадрата целиком, а хранятся только соответствующие функции.

Особенностью используемых на практике алгоритмов является их фиксированность, т. е. отсутствие в них изменяемых параметров, которые бы позволяли строить широкие классы латинских квадратов. Для случая множества n-мерных строк над полем $\mathbb{F}_2$ алгоритмы построения латинских квадратов, содержащие параметры и допускающие возможность их изменения в широком диапазоне были предложены в работе [1]. В данной работе рассматривается случай множества n-мерных строк над простым полем $\mathbb{F}_p$ и для данного множества предлагается соответствующая конструкция.

2. Пусть $\mathbb{F}_p$ — поле вычетов по модулю простого числа p, $\mathbb{F}_p^n$ — множество n-мерных строк над полем $\mathbb{F}_p$. Всякий латинский квадрат над множеством $\mathbb{F}_p^n$ может быть задан системой n функций p-значной логики от $2n$ переменных:

$$\begin{array}{l} f_1(x_1, \ldots, x_n, y_1, \ldots, y_n) \\ f_2(x_1, \ldots, x_n, y_1, \ldots, y_n) \\ \ldots \\ f_n(x_1, \ldots, x_n, y_1, \ldots, y_n), \end{array} \tag{3}$$

где набор $(x_1, \ldots, x_n)$ задает номер строки, $(y_1, \ldots, y_n)$ — номер столбца, соответствующие значения функций $(f_1, \ldots, f_n)$ определяют элемент квадрата.

Справедлива

Теорема 1. *Семейство функций p-значной логики $f = (f_1, \dots, f_n)$ от $2n$ переменных $x_1, \dots, x_n, y_1, \dots, y_n$ определяет латинский квадрат тогда и только тогда, когда во всех произведениях $f_{i_1}^{\alpha_1} \dots f_{i_k}^{\alpha_k}$, кроме $f_1^{p-1} \dots f_k^{p-1}$, где $1 \leqslant i_1 < \dots < i_k \leqslant n$, $1 \leqslant \alpha_i \leqslant p-1$, $i = 1, \dots, k$, $1 \leqslant k \leqslant n$, коэффициенты при членах $x_1^{p-1} \dots x_n^{p-1}$ и $y_1^{p-1} \dots y_n^{p-1}$ в приведенных многочленах равны 0, а в произведении $f_1^{p-1} \dots f_n^{p-1}$ соответствующие коэффициенты равны 1.*

3. Предложим следующую конструкцию, которая позволяет более эффективно решать поставленные вопросы.

Пусть задано семейство функций p-значной логики $g = (g_1, \dots, g_n)$ от переменных $z_1, \dots, z_n$. Пусть $\pi_1(x_1, y_1), \dots, \pi_n(x_n, y_n)$ — система функций p-значной логики от 2-х переменных. Определим семейство функций p-значной логики $f_1, \dots, f_n$ от переменных $x_1, \dots, x_n, y_1, \dots, y_n$ соотношениями:

$$\begin{aligned} f_1 &= H_1(x_1, y_1, g_1(\pi_1(x_1, y_1), \dots, \pi_n(x_n, y_n))) \\ f_2 &= H_2(x_2, y_2, g_2(\pi_1(x_1, y_1), \dots, \pi_n(x_n, y_n))) \\ &\dots \\ f_n &= H_n(x_n, y_n, g_n(\pi_1(x_1, y_1), \dots, \pi_n(x_n, y_n))), \end{aligned} \tag{4}$$

где H_i, $i \in \overline{1,n}$ — функции p-значной логики от 3-х переменных.

Напомним (см. [1]), что семейство функций $g = (g_1, \dots, g_n)$ от переменных $z_1, \dots, z_n$ называется правильным, если для любых различных наборов $z' = (z'_1, \dots, z'_n)$ и $z'' = (z''_1, \dots, z''_n)$ существует $\alpha \in \overline{1,n}$, такое, что выполнено

$$z'_\alpha \neq z''_\alpha \quad \text{и} \quad g_\alpha(z') = g_\alpha(z''). \tag{5}$$

Пусть функции H_i, $i \in \overline{1,n}$ удовлетворяют условиям:

В уравнении

$$H(x, y, z) = t$$

над F_p при любых фиксированных трех величинах однозначно определена четвертая. Данное свойство для краткости будем называть латинским свойством.

Справедлива

Теорема 2. *Для латинских функций трех переменных H_i, $i \in \overline{1,n}$ семейство функций $f = (f_1, \dots, f_n)$ от $2n$ переменных вида (4) определяет латинский квадрат при любых функциях $\pi_1, \dots, \pi_n$ в том и только том случае, когда семейство функций $g = (g_1, \dots, g_n)$ является правильным.*

4. Рассмотрим вопрос о выполнении латинского свойства на функции H_i, $i \in \overline{i,n}$. Ясно, что им обладают линейные функции, зависящие от всех трех переменных над $\mathbb{F}_p$. Нетрудно доказать, что таковыми будут, в частности, все функции вида $\varphi_1(x) + \varphi_2(y) + \varphi_3(z)$, где φ_i — перестановочные многочлены над $\mathbb{F}_p$. Имеются классы перестановочных многочленов и их классификация для малых степеней и малых p (см. [3]).

5. Рассмотрим теперь вопрос о выполнении условий правильности для семейств функций p-значной логики. Отметим сначала, что условие правильности семейства функций может быть сведено к условию регулярности в следующем смысле.

Справедлива

Теорема 3. *Функции p-значной логики $g = (g_1, \dots, g_n)$ от переменных $x_1, \dots, x_n$ образуют правильное семейство тогда и только тогда, когда для любых наборов $a = (a_1, \dots, a_n)$ из $\mathbb{F}_p^n$ семейство $g(a) = (x_1 + a_1 g_1, \dots, x_n + a_n g_n)$ является регулярным.*

Следствие 1. *Семейство функций p-значной логики $g = (g_1, \dots, g_n)$ правильно тогда и только тогда, когда для любых двух наборов $a = (a_1, \dots, a_n) \in \mathbb{F}_p^n$ и $t = (t_1, \dots, t_n) \in \mathbb{F}_p^n$, причем $(t_1, \dots, t_n) \neq (0, \dots, 0)$ функция $t_1 x_1 + \dots + t_n x_n + t_1 a_1 g_1 + \dots + t_n a_n g_n$ имеет равномерный обобщенный вес.*

6. Для семейства функций $f = (f_1, \ldots, f_n)$ от переменных $x_1, \ldots, x_n$ определим граф существенной зависимости $G_f = (V, E)$, где $V = \{1, 2, \ldots, n\}$. Пара $(i, j) \in E \Leftrightarrow$ если f_j зависит от x_i существенно. Рассмотрим вопрос, как влияют циклы графа G_f семейства функций f на правильность этого семейства.

Теорема 4. *Пусть $f = (f_1, \ldots, f_n)$ — семейство функций p-значной логики, G_f — его граф существенной зависимости переменных. Пусть для любого простого элементарного цикла C графа G_f выполнено*

$$\prod_{i \in C} f_i \equiv 0. \tag{6}$$

Тогда семейство f является правильным.

Будем говорить, что функция $f(x_1, \ldots, x_n)$ p-значной логики обладает свойством Q, если для всякого существенного переменного x_k функции f выполнено: если $f(\alpha_1, \ldots, \alpha_k, \ldots, \alpha_n) \neq 0$, то существует $\alpha'_k \in \mathbb{F}_p$, такое, что $f(\alpha_1, \ldots, \alpha'_k, \ldots, \alpha_n) = 0$. (Т. е. на каждом ребре p-ичного куба есть нуль функции.)

Теорема 5. *Пусть каждая функция семейства $f = (f_1, \ldots, f_n)$ функций p-значной логики обладает свойством Q. Тогда для правильности семейства f необходимо, чтобы выполнялось условие* (6) *для каждого простого элементарного цикла графа G_f.*

Литература

[1] Носов В. А. Построение классов латинских квадратов в булевой базе данных. Интелл. Системы, т. 4, вып. 3–4, 1999, с. 307–320.

[2] Применко В. А., Скворцов В. Ф. Об условиях регулярности конечных автономных автоматов. Дискретная математика, т. 2, вып. 1, 1990.

[3] Лидл Р., Нидеррайтер Т. Конечные поля, т. 2. М.: Мир, 1968.

[4] Шеннон К. Теория связи в секретных системах. Работы по теории информации и кибернетике. М.: 1963, с. 333–369.

[5] Denes J., Keedwell A. Latin aquares and their applications. Budapest, 1974.

Программная реализация генерации серий бинарных многочленов[15]

О. М. Баданова, А. В. Усольцев

Работа посвящена программным реализациям алгоритмов генерации многочленов с данными свойствами (над полем $\mathbb{Z}_2$), в том числе «криптографических» многочленов [1]. Осуществлено построение таблицы степеней порождающего элемента по модулю данного многочлена, что в случае, когда многочлен примитивен, дает представление соответствующего конечного поля характеристики два [2]. Это дает возможность построения линейных регистров сдвига с данным периодом, линейных рекуррентных последовательностей [2, 3], используемых в при гаммировании (возможно, с блоками усложнения [4]),

Для решения задачи получения многочлена заданного порядка (обычно предполагаемого простым) предложена программа нахождения наибольшего общего делителя соответствующей геометрической прогрессии и суммы степеней переменной по всем показателям, являющимся степенями двойки [6].

[15] Работа поддержана грантом РФФИ № 01-01-00688.

Входными данными является P – порядок искомого многочлена. В результате своей работы программа находит многочлен как наибольший общий делитель и его степень. Так, например, при $P = 1163$ программа находит неприводимый многочлен степени $n = 166$. Размер откомпилированной программы 309 Кб, в исходном тексте программы 210 строк.

Предложена также программа для нахождения серии (рекуррентной последовательности) неприводимых многочленов данной степени по имеющемуся исходному неприводимому многочлену. Построение производится с использованием рекуррентных формул перехода от многочлена $f(x)$ с корнем α, $f(\alpha) = 0$, к многочлену $g(x)$ с корнем α^p, $g(\alpha^p) = 0$, при $p = 3$ или $p = 5$ [7]. Начальные данные – коэффициенты известного неприводимого многочлена $f(x)$ с корнем α.

При $p = 3$ размер откомпилированной программы 380 Кб, в исходном тексте программы 160 строк.

При $p = 5$ размер откомпилированной программы 386 Кб, в исходном тексте программы 168 строк.

Этот подход позволяет при получении неприводимого многочлена порядка P степени n по алгоритму Евклида в случае выполнения простых теоретико-числовых соотношений (например, если тройка или пятерка являются первообразными корнями по модулю P) автоматически сгенерировать все неприводимые многочлены порядка P степени n, в частности, для выработки кода помехоустойчивой передачи данных [8].

Особенно эффективно применение этой программы для «криптографических» многочленов и линейных рекуррентных последовательностей максимального периода, в том случае, когда $2^n - 1$ – простое число Мерсенна [1, 3, 5].

Литература

[1] Шнайер Б. Прикладная криптография. 2-е издание: протоколы, алгоритмы и исходные тексты на языке С. 1996. (пер. с англ.: Bruce Schneier. Applied Cryptography. Second Edition: Protocols, Algorithms, and Source Codes in C. John Wiley & Sons. 1996. 758 p.)

[2] Лидл Р., Нидеррайтер Г. Конечные поля. М.: Мир. 1988.

[3] Бабаш А. В., Шанкин Г. П. Криптография. Под редакцией И. П. Шерстюка, Э. А. Применко / Серия книг «Аспекты защиты». М.: СОЛОН-Р. 2002. 512 с.

[4] Масленников М. Е. Практическая криптография. СПб: БХВ-Петербург. 2003. 464 с. + CD.

[5] Бухштаб А. А. Теория чисел. М.: Просвещение. 1966. 384 с.

[6] Баданова О. М., Иициксон М. А., Титов С. С., Усольцев А. В. Вычисление коэффициентов неприводимых делителей суммы геометрической прогресии Проблемы теоретической и прикладной математики. Труды 34-й Региональной молодежной конференции. Екатеринбург: УрО РАН. 2003. С. 3–4.

[7] Демкина О. Е., Титов С. С., Торгашова А. В. Рекуррентное вычисление коэффициентов степеней экспоненты Проблемы теоретической и прикладной математики. Труды 34-й Региональной молодежной конференции. Екатеринбург: УрО РАН. 2003. С. 27–30.

[8] Яковлев В. В., Корниенко А. А. Информационная безопасность и защита информации в корпоративных сетях железнодорожного транспорта. Учебник для вузов ж.-д. транспорта / Под ред. В. В. Яковлева. М.: УМК МПС России. 2002. 328 с.

Построение негрупповых латинских квадратов произвольно больших порядков

Л. Э. Будагян

Латинский квадрат (сокращение *ЛК*) порядка m — это квадратная матрица размера m, заполненная элементами множества мощности m таким образом, что ни в одной строке и ни в одном столбце нет совпадающих элементов.

Латинские квадраты имеют множество применений. Так, например, в известной работе К. Шеннона «Теория связи в секретных системах» [7] приводится пример того, как можно использовать ЛК для построения так называемых «совершенно секретных систем». ЛК также применяются для построения кодов, исправляющих ошибки, в теории планирования эксперимента. Кроме того, латинский квадрат как математический объект тесно связан с группами.

Условие четырехугольника (quadrangle criterion) для латинских квадратов:

Говорят, что для ЛК L выполняется условие четырехугольника, если для любых индексов i_1, j_1, i'_1, j'_1, i_2, j_2, i'_2, j'_2 из условий

$$\begin{aligned} L(i_1, j_1) &= L(i_2, j_2), \\ L(i'_1, j'_1) &= L(i'_2, j'_2), \\ L(i'_1, j_1) &= L(i'_2, j_2) \end{aligned}$$

следует выполнение условия

$$L(i_1, j'_1) = L(i_2, j'_2).$$

Приведем два факта, показывающих связь ЛК с группами (позаимствованы из [1]):

Теорема 1. *Таблица Кэли (таблица умножения) конечной группы G есть ЛК, для которого выполняется условие четырехугольника, и наоборот, всякий ЛК, удовлетворяющий условию четырехугольника, можно окаймить таким образом, что он станет таблицей Кэли некоторой группы.*

Теорема 2. *На множестве ЛК порядка m можно ввести трехуровневую классификацию: $i.j.k$, где i — номер главного класса, j — номер класса изотопии в i-ом главном классе, k — номер класса изоморфизма в $i.j$-ом классе изотопии. Если главный класс содержит групповой ЛК (т. е. удовлетворяющие условию четырехугольника), то он является классом изотопии, а также изоморфизма.*

Для решения задачи построения ЛК высоких порядков используем конструкцию над полем $\mathbb{Z}_p$, p — простое. Для этого понадобятся следующие утверждения:

Утверждение 1. *Формула $L(x, y) = \pi(x + y) + x$, где сложения используются в смысле $\mathbb{Z}_p$, задает ЛК $\iff$ $\pi(x) \in S_p, \sigma(x) = \pi(x) + x \in S_p$, где S_p — симметрическая группа (группа перестановок) порядка p.*

Утверждение 2. *Если $L(x, y) = \pi(x + y) + x$ задает ЛК, то то же самое будет верно и для всякой перестановки, полученной из π заменой в ее цикловой структуре части циклов на обратные им.*

Кроме того, очевидно, что для всякого нечетного числа p формула $k(x + y) + x$, $k \neq -1$ над $\mathbb{Z}_p$ задает ЛК.

Для всякого разложения $p - 1 = l \cdot m$, $l > 2, m \geqslant 2$ по элементу k мультипликативной группы $\mathbb{Z}_p^*$ порядка l (всего их $\varphi(l)$; $k \neq \pm 1$ т. к. $ord(k) = l > 2$) можно построить 2^m перестановок, задающих ЛК по формуле из утверждения 1 — по одной для каждого *дескриптора* $\varepsilon = (\varepsilon_0, \ldots, \varepsilon_{m-1})$, $\varepsilon_i = \pm 1$ — следующего вида:

$$\begin{aligned} \pi_\varepsilon(0) &= 0; \\ \pi_\varepsilon(x) &= k^{\varepsilon_i}, \quad x \in C_i, \end{aligned}$$

где используется обозначение: $C_i = \{s^i \cdot k^j\}_{j=0}^{l-1}$.

Для этих построений удалось получить следующий результат:

Теорема 3. *Если $\varepsilon \neq (1, \ldots 1)$, то для ЛК $L(x, y) = \pi_\varepsilon(x + y) + x$ не выполняется условие четырехугольника.*

Таким образом, подавляющее большинство полученных ЛК представляет собой негрупповые ЛК, что является доказательством их неизоморфности таблице умножения циклической группы соответствующего порядка и производным из нее латинским квадратам (см. теоремы 1, 2).

Построенные таким образом ЛК имеют следующие преимущества:

- Указанный алгоритм позволяет строить негрупповые ЛК для произвольного простого порядка, большего 5, то есть имеется новое семейство ЛК, содержащее произвольно большие ЛК. Построение ЛК высоких порядков является нетривиальной задачей.
- Для хранения и передачи построенных ЛК требуется запомнить лишь порождающую перестановку π, которая, в свою очередь, порождается несколькими параметрами. Это важно для многих применений ЛК, в том числе криптографических.
- Можно показать, что сложность построения ЛК подобным образом по этим параметрам *линейно* зависит от порядка ЛК.
- Для построенных ЛК $z = L(x, y)$ можно легко построить обратный: $y = L^{-1}(x, z)$. Это обстоятельство также имеет большое значения для многих их применений (в криптографии, теории кодирования).

Литература

[1] Dénes J., Keedwell A. D. Latin Squares and their Applications. Budapest: Akadémiai Kiadó, 1974.

[2] Белоусов В. Д., Белявская Г. Б. Латинские квадраты, квазигруппы и их приложения. Кишинев: Штиинца, 1989.

[3] Носов В. А. О построении классов латинских квадратов в булевой базе данных. Интеллектуальные системы, т. 4, вып. 3–4, 1999.

[4] Лидл Л., Нидеррайтер Г. Конечные поля. В 2-х т. М.: Мир, 1988.

[5] Кострикин А. И. Введение в алгебру. В 3-х частях. М.: Физико-математическая литература, 2000.

[6] Виноградов И. М. Основы теории чисел. М.: Наука, Главная редакция физико-математической литературы, 1981.

[7] Шеннон К. Э. Работы по теории информации и кибернетике. М.: Издательство иностранной литературы, 1963.

[8] Холл М. Комбинаторика. М.: Мир, 1970.

Автоматная модель описания динамики вторжения

С. С. Корт

В тезисах доклада представлена модель описания динамики вторжения для системы обнаружения вторжений, основанная на конечном автомате, описывающем динамику развития вторжения. Под вторжением будем понимать нарушение безопасности, состоящее из одной или нескольких атак. Типовой сценарий вторжения в общем случае состоит из следующих этапов.

1. Этап сбора информации. На этапе сбора информации нарушителя может интересовать информация об атакуемой системе, в том числе:

- топология сети, в которой функционирует атакуемая система;
- тип ОС на атакуемых хостах;
- функционирующие на хостах сервисы;
- дополнительная информация об атакуемых хостах.

2. Этап непосредственной атаки. На основании информации об атакуемом объекте, собранной в результате выполнения предыдущего этапа, нарушитель может начать атаку на систему. На данном этапе используются типовые уязвимости в системных сервисах или ошибки в администрировании системы. Успешным результатом использования уязвимостей обычно является получение нарушителем прав на атакованном хосте, получение файла паролей, отказ в обслуживании атакуемого хоста и т. д.

3. Этап консолидации. После того как нарушитель осуществил с использованием атаки на систему проникновение в нее, обычно начинается использование скомпрометированного хоста - этап консолидации. Данная стадия вторжения может быть подразделена на две логические фазы: консолидация и распространение вторжения. Кроме того, к данной фазе можно отнести и действия нарушителя, связанные с реализацией угроз безопасности (например, доступ к защищаемой информации).

Существующие методы обнаружения вторжений можно в общем случае охарактеризовать следующим образом:

- сигнатурные методы выявления атак направлены на выявление составных элементов вторжения и не пытаются объединить полученные результаты в единую картину вторжения;
- существующие методы искусственного интеллекта, дополняющие сигнатурные методы выявления атак также не учитывают этапы вторжения;
- методы выявления аномалий исследуют отклонения в поведении пользователей и не позволяют отследить картину вторжения.

В основу алгоритма анализа динамики развития вторжения заложены сценарии вторжения, основанные на взаимосвязи различных этапов вторжения. Этот алгоритм анализирует состояние внешних хостов сети по отношению к защищаемому. При этом внешний хост может быть охарактеризован следующим образом:

1. Неизвестный хост - хост, обращавшийся к защищаемому хосту в течение краткого периода времени, и не производивший атакующих действий;
2. Доверенный хост - хост, обращавшийся к защищаемому хосту в течение длительного периода времени, и не производивший атакующих действий;
3. Подозрительный хост - хост, сканировавший или атаковавший защищаемый хост.

Данные для анализа с использованием данного метода, управляющие переходами автомата, поступают от системы обнаружения атак Snort.

Отношения «внешний хост» - «защищаемый хост» описываются в виде автомата конечных состояний (показан на рисунке 1).

Переходы данного автомата можно описать выражением:

Transition (Action, ServiceName, StandartRecone, OSDepended, UserDepended, FSDepended, UsedTool)

В описании перехода автомата определены следующие переменные:

1. Action - описывает действие нарушителя; возможны следующие типы действий:
   a) сканирование - сервиса (ReconService), операционной системы (ReconOS), информации о файловой системе хоста (ReconFS), учетных записях пользователей (ReconUser); одно и то же сканирование может одновременно относится к нескольким типам;
   b) атаки отказа в обслуживании - сервиса (DOSService), хоста (DosHost); атаки на учетную запись - администратора (ServiceAdminAttack), пользователя (ServiceUserAttack), атаки чтения объекта файловой системы (ReadFile), записи в объект файловой системы (WriteFile); каждая атака может быть только одного из перечисленных типов;
   c) консолидации - с использованием известного троянского ПО (ConsolidationTroyan), выполнение команд на защищаемом сервере (Consolidation).

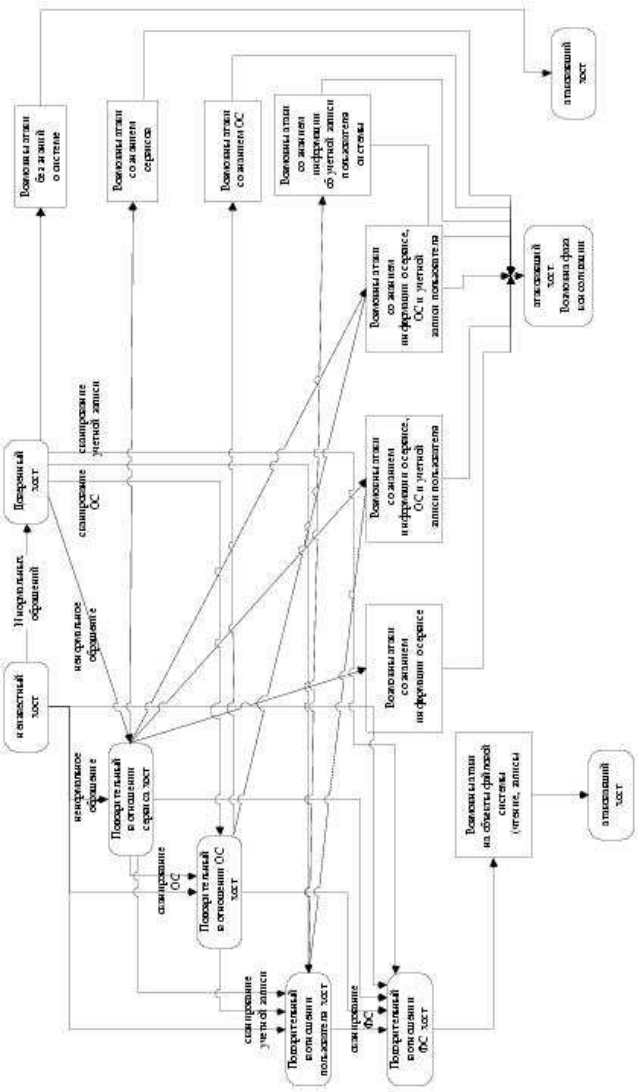


Рис. 1: Автомат, описывающий сценарий атак.

2. ServiceName - имя сервиса используемого нарушителем для атаки; в том случае, если действие нарушителя не зависит от конкретного сервиса, используется ключевое слово «MISC».

3. StandartRecone - использование стандартных методов обращения к защищаемому хосту; может принимать значения «Yes», «No».

4. OSDepended - зависимость фазы атаки от знаний об операционной системе, установленной на защищаемом хосте; может принимать значения «Yes», «No».

5. UsedTool - использование известных средств нападения может принимать значения «Yes», «No».

Для каждого сообщения, классифицированного как сообщение об атаке или фазе консолидации, производится верификация сценария в соответствии с текущим состоянием обращающегося хоста по отношению к сервису, ОС, подсистеме учетных записей и файловой системе защищаемого хоста. Верификация означает сопоставление атрибутов «подозрительности» обращающегося хоста с атрибутами, описывающими классифицированное действие.

Признаком успешного (верифицированного) вторжения является обнаружение сигнатуры, соответствующей этапу непосредственной атаки, после обнаружения сигнатуры сканирования с нужными атрибутами.

Признаком успешного (верифицированного) вторжения (конечное состояние автомата) является обнаружение сигнатуры, соответствующей этапу консолидации после обнаружения сигнатуры непосредственной атаки нужного типа.

Таким образом, разработанный метод анализа динамики развития вторжения позволяет:

- определить атаки возможные на сервис в соответствии со сценарием (предсказание сценария вторжения);

- оценить сценарий развития вторжения (верификация сценария вторжения).

Обратимые клеточные автоматы

И. В. Кучеренко

Клеточный автомат (КА) представляет собой бесконечную однородную сеть автоматов, которая может интерпретироваться как дискретная динамическая система. Для приложений в области защиты информации особый интерес представляют обратимые клеточные автоматы. Укажем некоторые результаты исследования свойств этого класса объектов.

k-конфигурацией клеточного автомата называется его конфигурация с конечным числом ячеек в состояниях, отличных от состояния покоя. Клеточный автомат называется обратимым, если каждая его k-конфигурация имеет не более одного прообраза, являющегося k-конфигурацией, и сильно обратимым, если у любой его конфигурации имеется ровно один прообраз. Каждый сильно обратимый КА является обратимым, но не наоборот. Отображение пространства состояний, обратное к отображению, реализуемому сильно обратимым клеточным автоматом, также является КА [4]. Следовательно, множество сильно обратимых КА образует группу относительно операции композиции.

Одним из наиболее важных параметров класса обратимых клеточных автоматов, существенно влияющим на его свойства, является размерность пространства ячеек. В процессе ее роста «устройство» обратимых КА сильно усложняется. Как выяснилось, с алгоритмической точки зрения наиболее значимым является переход от размерности 1 к 2. Для одномерных КА имеются алгоритмы проверки свойств обратимости и сильной обратимости, имеющие квадратичную сложность [8]. В двумерном случае эти свойства алгоритмически не распознаваемы [6]. Более того, проверка свойства необратимости на k-конфигурациях ограниченного размера является NP-полной задачей [7].

Хорошо известно, что доля обратимых клеточных автоматов среди всех КА с заданными числом состояний ячейки и шаблоном соседства стремится к нулю с ростом числа состояний или числа векторов в шаблоне соседства [1]. Тем не менее, этот класс является достаточно богатым: конструктивно

строится подкласс класса обратимых КА, асимптотика логарифма числа элементов в котором совпадает с асимптотикой логарифма числа всех клеточных автоматов [3]. Кроме того, оба класса обратимых КА допускают вложения в них любых клеточных автоматов меньшей размерности. Автором выделен минимальный класс обратимых КА, достаточный для представления любого клеточного автомата с фиксированными шаблоном соседства и числом состояний ячейки n. Вложение в этот класс может быть проведено конструктивно. При этом число векторов в шаблоне соседства моделирующего КА увеличивается на единицу по сравнению с числом векторов у вкладываемого, а число состояний ячейки остается неизменным.

Возможность представления произвольного КА в сильно обратимом большей размерности впервые была продемонстрирована в работе [5]. Вложение может быть осуществлено в класс, имеющий n^2 состояний ячейки, размерность пространства ячеек, на единицу превосходящую размерность вкладываемого КА, и один «дополнительный» вектор в шаблоне соседства [9]. В общем случае «упростить» класс моделирующих клеточных автоматов за счет уменьшения размерности пространства ячеек оказывается невозможным из-за того, что никакой необратимый КА нельзя вложить в сильно обратимый клеточный автомат той же размерности [9].

Литература

[1] Кудрявцев В. Б., Подколзин А. С., Болотов А. А. Основы теории однородных структур. М.: Наука, 1990.

[2] Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. М.: Наука, 1985.

[3] Кучеренко И. В. О числе обратимых однородных структур. Дискретная математика, в печати.

[4] Richardson D. Tesselations with Local Transformations. Journal of Computer and System Sciences, 6: 373–388, 1972.

[5] Toffoli T. Computation and Construction Universality of Reversible Cellular Automata. Journal of Computer and System Sciences, 15: 213–231, 1977.

[6] Kari J. Reversibility and Surjectivity Problems of Cellular Automata. Journal of Computer and System Sciences, 48(1): 149–182, 1994.

[7] Durand B. Inversion of 2D cellular automata: some complexity results. Theoretical Computer Science, 134(2): 387–401, 1994.

[8] Sutner K. Linear Cellular Automata and De Bruijn Automata. In: Cellular Automata: a parallel model, (eds. M. Delorme and J. Mazoyer), Kluwer: 303–319, 1998.

[9] Hertling P. Embedding Cellular Automata into Reversible Ones. In: Unconventional Models of Computation (eds. C. S. Claude, J. Casti and M. J. Dinneen), Springer-Verlag, 243–256, 1998.

Решение автоматных уравнений[16]

И. В. Лялин

Пусть S — автоматная схема, в которой один из автоматов обозначен как x. Будем называть x свободной позицией. Если подставить в S вместо x какой-то другой автомат f, то схема бует реализовать какой-то автомат h. Будем записывать это так: $S(f) = h$.

Задача: пусть даны автоматная схема S со свободной позицией x и автомат h. Требуется найти такой автомат f, чтобы $S(f) = h$. Иными словами, решить уравнение $S(x) = h$.

Варианты задачи:

[16]Работа выполнена при финансовой поддержке РФФИ, проект 00-01-00374.

1. Определить имеет ли уравнение решение.

2. Как-то описать множество решений заданного автоматного уравнения.

Обе задачи были эффективно решены.

Пусть $k \geqslant 2$, $t \geqslant 1$, E_k^t — множество всех слов длины t, а E_k^∞ — множество всех бесконечных последовательностей, составленных из элементов $E_k = \{0, 1, \ldots, k-1\}$. Через M обозначим множество всех подмножеств множества E_k^∞.

Пусть $g(x_1, \ldots, x_n)$ — недетерминированная функция, отображающая множество $E_k^\infty \times \ldots \times E_k^\infty$ в M. Множество значений, которые $g(x_1, \ldots, x_n)$ принимает на наборе $(\alpha_1, \ldots, \alpha_n)$, принадлежащего либо $E_k^\infty \times \ldots \times E_k^\infty$, либо для некоторого $t \geqslant 1$ $E_k^t \times \ldots \times E_k^t$, обозначим $\{g(\alpha_1, \ldots, \alpha_n)\}$ или $\{g(\alpha_1, \ldots, \alpha_n)\}^t$ соответственно. По аналогии с определением о.-д. функции будем считать, что недетерминированная функция является ограниченно-недетерминированной (о.-н. д. функцией), если число «остаточных» функций для g конечно.

О.-д. функция $f(x_1, \ldots, x_n)$ по определению «вложима» в о.-н. д. функцию $g(x_1, \ldots, x_n)$ тогда и только тогда когда для любых $\alpha_1 \in E_k^\infty, \ldots, \alpha_n \in E_k^\infty$ $f(\alpha_1, \ldots, \alpha_n) \in \{g(\alpha_1, \ldots, \alpha_n)\}$.

Пусть $g(x_1, \ldots, x_n)$ — произвольная о.-н. д. функция. Через N_g обозначим множество всех тех и только тех о.-д. функций, которые «вложимы» в g. Нетрудно видеть, что множество N_g перечислимо.

Пусть $S(x) = h$ — произвольное автоматное уравнение. Имеет место следующая теорема:

Теорема 1. *Существует алгоритм, определяющий существует или нет решение уравнения* $S(x) = h$.

Теорема 2. *Если уравнение* $S(x) = h$ *имеет решение то эффективно строится о.-н. д. функция* g *такая, что множество* N_g *совпадает с множеством всех решений данного уравнения.*

Литература

[1] Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Элементы теории автоматов. Издательство Московского Университета, 1978, Глава 3, § 1.

[2] Petrenko A., Yevtushenko N. Solving asuynchronous equations. Formal description techniques // Protocol specification, testing and verification. Kluwer Academic Publishers, 1998, p. 125–140.

Вероятностные модели и статистическое тестирование случайных и псевдослучайных последовательностей

Ю. С. Харин

1 Введение

Случайные и псевдослучайные последовательности являются неотъемлемыми элементами криптосистем [1, 2, 3, 4]: гамма в поточных криптосистемах; сеансовые и другие ключи для блочных криптосистем; стартовые значения для генерации ряда математических величин в асимметричных криптосистемах, например «больших простых чисел» в криптосистемах RSA, ElGamal; случайные значения параметров для многих систем ЭЦП, например DSA, СТБ 1176.2-99; случайные выборы в протоколах аутентификации, например в протоколах Cerberos, SET, SSL. Для обеспечения требуемой стойкости криптосистем генерируемые последовательности по своим свойствам «должны приближаться к свойствам равномерно распределенной случайной последовательности» (РРСП) или, как ее часто называют в криптографических приложениях, «чисто случайной» последовательности. В докладе дается обзор

существующих методов тестирования последовательностей и предлагается новый подход к построению тестов на основе моделей дискретных временных рядов.

2 РРСП и ее вероятностные свойства

РРСП — это случайная последовательность $x_1, x_2, \ldots, x_t, x_{t+1}, \ldots \in \mathcal{A}$ со значениями в множестве $\mathcal{A} = \{0, 1, \ldots, N-1\}$, $N \geqslant 2$, определенная на вероятностном пространстве $(\Omega, \mathcal{F}, P)$ и удовлетворяющая двум свойствам:

C1. $\forall\, n \in \mathbb{N}$ и произвольных значений индексов $1 \leqslant t_1 < \ldots < t_n$ случайные величины $x_{t_1}, \ldots, x_{t_n} \in \mathcal{A}$ независимы в совокупности.

C2. $\forall\, t \in \mathbb{N}$ случайная величина x_t имеет дискретное равномерное на $\mathcal{A}$ распределение вероятностей: $\mathsf{P}\{x_t = i\} = N^{-1}$, $i \in \mathcal{A}$.

Из базовых свойств C1, C2 вытекают следующие дополнительные свойства, часто используемые при статистическом тестировании последовательностей:

C3. Если $\{x_t\}$ — РРСП, то $\forall\, n \in \mathbb{N}$, $1 \leqslant t_1 < \ldots < t_n$ n-мерное распределение слова $(x_{t_1}, \ldots, x_{t_n}) \in \mathcal{A}^n$ является равномерным:

$$\mathsf{P}\{x_{t_1} = i_1, \ldots, x_{t_n} = i_n\} = p^0_{i_1,\ldots,i_n} = N^{-n}, \quad i_1, \ldots, i_n \in \mathcal{A}. \tag{1}$$

C4. Если $\{x_t\}$ — РРСП, то для начального момента k-го порядка ($k \in \mathbb{N}$), ковариационной функции и спектральной плотности справедливы следующие формулы:

$$\alpha_k = \mathsf{E}\{x^k\} = \frac{1}{N(k+1)} \sum_{l=0}^{k} \binom{k+1}{l} B_l N^{k+1-l},$$

$$r(\tau) = \mathsf{E}\{(x_t - \alpha_1)(x_{t+\tau} - \alpha_1)\} = \delta_{\tau,0} \frac{N^2 - 1}{12},$$

$$\tau \in \mathbb{Z}, \quad S(\lambda) = \frac{N^2 - 1}{24\pi}, \quad \lambda \in [-\pi, \pi],$$

где $\{B_l\}$ — числа Бернулли, δ_{ij} — символ Кронеккера.

C5. (Воспроизводимость при прореживании.) Для любой фиксированной последовательности $1 \leqslant t_1 < \ldots < t_n < \ldots$ при «прореживании» РРСП $\{x_t\}$ возникает подпоследовательность $y_1 = x_{t_1}, \ldots, y_n = x_{t_n}, \ldots$, которая также является РРСП.

C6. (Воспроизводимость при суммировании.) Если $\{x_t\}$ — РРСП, а $\{\xi_t\}$ — произвольная неслучайная последовательность, либо случайная последовательность, не зависящая от $\{x_t\}$, то последовательность $y_t = (x_t + \xi_t) \bmod N$ также является РРСП.

C7. Если $\{x_t\}$ — РРСП, то $\forall\, n \in \mathbb{N}$ количество информации по Шеннону, содержащейся в слове $X_n = (x_1, \ldots, x_n) \in \mathcal{A}^n$ о будущем символе x_{n+1}, равно нулю: $\mathcal{J}\{x_{n+1}, X_n\} = 0$, поэтому для любого алгоритма прогнозирования $\hat{x}_{n+1} = f(X_n)$ вероятность ошибки прогнозирования не может быть сделана меньше, чем для «угадывания по жребию»: $\mathsf{P}\{\hat{x}_{n+1} \neq x_{n+1}\} = 1 - N^{-1}$.

3 Обзор существующих тестов РРСП

В [3] приведен обзор более 20 статистических тестов для проверки гипотез $H_0 = \{\{x_t\}$ есть РРСП$\}$, $H_1 = \overline{H}_0$, в том числе тестов из «батареи Кнута» [5], «батареи Марсальи «Diehard» [6], набора критериев NIST SP 800-22 [7], разработанного национальным институтом стандартизации США и использованного в конкурсе AES для оценивания качества выходных последовательностей алгоритмов блочного шифрования, а также набора критериев, использованного в конкурсе NESSIE [8]. Проведенный обзор позволяет сделать следующие выводы: 1) многие тесты ориентированы на проверку лишь одного из свойств C1–C7; 2) многие тесты построены «эвристически» и не фиксируют класс альтернатив, которые обнаруживаются и не обнаруживаются данными тестами; 3) многие тесты не имеют оценок мощности.

4 Построение тестов РРСП с использованием моделей дискретных временных рядов

Пусть наблюдается подлежащая тестированию реализация $X_n = (x_1, \ldots, x_n) \in \mathcal{A}^n$ случайной или псевдослучайной последовательности. Обозначим Π_n множество всевозможных дискретных вероятностных распределений на $\mathcal{A}^n$:

$$\Pi_n = \left\{ p = (p_{i_1,\ldots,i_n}) : 0 \leqslant p_{i_1,\ldots,i_n} \leqslant 1, \sum_{(i_1,\ldots,i_n)\in\mathcal{A}^n} p_{i_1,\ldots,i_n} = 1 \right\}.$$

Каждое вероятностное распределение $p = (p_{i_1,\ldots,i_n})$ однозначно определяется $M_n = N^n - 1$ элементарными вероятностями. Гипотезе H_0 в пространстве Π_n соответствует равномерное распределение (1). С целью уменьшения экспоненциально растущего числа параметров M_n рассмотрим m_n — параметрическое семейство вероятностных распределений на $\mathcal{A}^n$:

$$\mathcal{P}_n = \{p = (p_{i_1,\ldots,i_n}) \in \Pi_n : p_{i_1,\ldots,i_n} = q_{i_1,\ldots,i_n}(\theta),\ (i_1,\ldots,i_n) \in \mathcal{A}^n,\ \theta \in \Theta\},$$

где $\theta = (\theta_1, \ldots, \theta_{m_n}) \in \Theta \subseteq \mathbb{R}^{m_n}$ — m_n-мерный вектор параметров, а $\{q_{i_1,\ldots,i_n}(\cdot)\}$ — известные функции, причем $m_n < M_n$ и $p^0 \in \mathcal{P}_n$, т. е. существует подмножество $\Theta_0 \subset \Theta : \forall\, \theta \in \Theta_0$, $q_{i_1,\ldots,i_n}(\theta) = p^0_{i_1,\ldots,i_n}$, $(i_1,\ldots,i_n) \in \mathcal{A}^n$. Для обеспечения достаточной точности проверки гипотез H_0, H_1 в условиях регулярности, в силу информационного неравенства [9], будем предполагать следующую асимптотику: $m_n/n \to \lambda$, $0 < \lambda < 1$.

Каждое распределение $p \in \mathcal{P}_n$ определяет некоторую модель дискретного временного ряда. Представим модели дискретных временных рядов, которые позволяют строить статистические тесты проверки гипотез H_0, H_1.

1. *Однородная цепь Маркова s-того порядка* ($s \geqslant 1$) с матрицей вероятностей одношаговых переходов $P = (p_{j_1,\ldots,j_s;j_{s+1}})$ и стационарным начальным распределением вероятностей $\pi = (\pi_{j_1,\ldots,j_s})$, $j_1, \ldots, j_{s+1} \in \mathcal{A}$. При этом θ — это вектор элементарных вероятностей матрицы P; с учетом условия нормировки число параметров $m_n = N^s(N-1)$.

2. *Модель Рафтери.* Для уменьшения числа параметров однородной цепи Маркова s-того порядка А. Рафтери в 1985 г. предложил [10] «малопараметрическую» модель, в которой матрица P вероятностей одношаговых переходов задается соотношением:

$$p_{i_1,\ldots,i_s;i_{s+1}} = \sum_{j=1}^{s} \lambda_j q_{i_j,i_{s+1}}, \quad i_1, \ldots, i_{s+1} \in \mathcal{A}, \tag{2}$$

где $Q = (q_{ij})$, $i, j \in \mathcal{A}$ — некоторая стохастическая матрица, а $\lambda = (\lambda_j)$ — некоторый вектор элементарных вероятностей. При этом вектор θ состоит из элементов Q, λ; число параметров $m_n = N(N-1) + s - 1$ и с увеличением порядка s растет линейно, а не экспоненциально. Подпространство параметров Θ_0, соответствующее гипотезе H_0, имеет вид: $\Theta_0 = \{\theta = (vec(Q)\vdots\lambda) : q_{ij} = N^{-1}, i, j \in \mathcal{A}\}$. Существует ряд обобщений модели (2).

3. *Дискретная авторегрессия порядка s: DAR(s).* Эта модель является модификацией известной для «непрерывных» временных рядов авторегрессионной модели AR(s) и задается стохастическим разностным уравнением s-того порядка над полем $GF(N)$:

$$x_t = \alpha_s x_{t-1} + \ldots + \alpha_1 x_{t-s} + \xi_t, \quad t = s+1, s+2, \ldots, \tag{3}$$

где $\alpha = (\alpha_1, \ldots, \alpha_s) \in \mathcal{A}^s$ — вектор коэффициентов авторегрессии ($\alpha_1 \neq 0$), $\{\xi_t\}$ — последовательность н. о. р. дискретных случайных величин с некоторым распределением вероятностей $\mathsf{P}\{\xi_t = j\} = \varepsilon_j$, $j \in \mathcal{A}$, $X_s = (x_1, \ldots, x_s) \in \mathcal{A}^s$ — случайный вектор начальных значений, не зависящий от $\{\xi_{s+1}, \xi_{s+2}, \ldots\}$ и имеющий некоторое фиксированное s-мерное распределение вероятностей. При этом θ состоит из элементов α, $\varepsilon = (\varepsilon_0, \ldots, \varepsilon_{N-1})$; число параметров $m_n = N + s - 2$; подпространство параметров $\Theta_0 =$

$\{\theta = (\alpha\vdots\varepsilon) : \varepsilon_j = N^{-1}, j \in \mathcal{A}\}$. Частными случаями модели (3) являются бинарная авторегрессия [11] при $N = 2$ и дискретная авторегрессионная модель Джекобса — Льюиса [12].

Использование моделей дискретных временных рядов позволяет построить тесты отношения правдоподобия для проверки H_0, H_1 и получить асимптотические оценки мощности для альтернатив, имеющих практическое значение в криптосистемах.

Литература

[1] Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В. Основы криптографии. М.: Гелиос, 2001.

[2] Харин Ю. С., Берник В. И., Матвеев Г. В. Математические основы криптологии. Мн.: БГУ, 1999.

[3] Харин Ю. С., Агиевич С. В. Компьютерный практикум по математическим методам защиты информации. Мн.: БГУ, 2001.

[4] Шнайер Б. Прикладная криптография. М.: Мир, 2001.

[5] Кнут Д. Э. Искусство программирования. Т. 2. М.: Вильямс, 2000.

[6] Marsaglia G. Keynote Address: A Current View of Random Number Generators // Proc. of the 16th Symp. "Computer Science and Statistics". N. Y.: Elsevier, 1985.

[7] NIST Special Publication 800–22: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. N. Y., 2000.

[8] NESSIE Report: List of General NESSIE Test Tools. (`http://www.cryptonessie.org`).

[9] Боровков А. А. Математическая статистика. М.: Наука, 2000.

[10] Raftery A. E. A Model for High-Order Markov Chains // Journal of the Royal Statistical Society, B. V. 47. No. 3. 1985. P. 528–539.

[11] Максимов Ю. И. О цепях Маркова, связанных с двоичными регистрами сдвига со случайными элементами. В кн.: Труды по дискретной математике. Т. 1. (Под ред. В. Н. Сачкова). М.: ТВиП, 1997. С. 203–220.

[12] Jacobs P. A., Lewis P. A. W. Autoregressive process DAR(p). Technical Report of Naval Postgraduate School. NPS 55-78-022, 1978.

Криптографические применения задачи о днях рождения

А. М. Зубков

Классическая «задача о днях рождения» рассматривается в большинстве учебников как эффектный неочевидный пример применения теории вероятностей. Она является частным случаем общего утверждения, состоящего в том, что если $\xi_1, \xi_2, \ldots, \xi_n$ – независимые случайные величины, имеющие равномерное распределение на множестве $\{1, \ldots, N\}$, то при $n, N \to \infty$, $n^2/N \to x \in (0, \infty)$ вероятность того, что все случайные величины $\xi_1, \xi_2, \ldots, \xi_n$ примут разные значения, стремится к $e^{-x/2}$, а распределение числа таких пар $(i, j) : 1 \leqslant i \leqslant j \leqslant n$, что $\xi_i = \xi_j$, стремится к распределению Пуассона с параметром $x/2$.

Это утверждение означает, что в последовательности независимых испытаний $\xi_1, \xi_2, \ldots$, результаты которых равномерно распределены на множестве $\{1, \ldots, N\}$, первое повторение исхода с вероятностью, близкой к 1, возникает после проведения $O\left(\sqrt{N}\right)$ испытаний.

В криптографическом анализе эффект, являющийся содержанием задачи о днях рождения, часто используется в немного другом виде. Пусть проведено две независимых серии по n независимых испытаний, результаты которых имеют равномерное распределение на множестве $\{1, \ldots, N\}$. Тогда при $N, n \to \infty, n^2/N \to x \in (0, \infty)$ вероятность того, что множества исходов, появившихся в первой и второй сериях испытаний, пересекаются, стремится к e^{-x}, а распределение размера пересечения сходится к распределению Пуассона с параметром x. Такое утверждение используется, в частности, для оценивания средней сложности метода, который на английском языке называют «meet-in-the-middle attack» (атака методом «встречи в середине»), и его вероятностных модификаций.

Исходный метод «встречи в середине» является переборным и детерминированным. Например, пусть шифрование блоков длиной m бит проводится в 2 этапа с помощью n-битовых ключей K_1 и K_2 по формуле

$$C = E^2_{K_2}(E^1_{K_1}(P)),$$

где P – блок открытого текста, C – блок шифрованного текста, E^1 и E^2 – операторы шифрования на первом и втором этапах, а D^1 и D^2 – обратные к E^1 и E^2 операторы.

Если известны две пары открытого и шифрованного текстов $(P_1, C_1), (P_2, C_2)$, то поиск секретных ключей методом «встречи в середине» (см., например, [1]) состоит в построении таблиц значений $E^1_k(P_1)$ для всех 2^n возможных значений k ключа первого блока и значений $D^1_k(C_1)$ для всех возможных значений k ключа второго блока. Каждое значение $E^1_{k_1}(P_1) = D^1_{k_2}(C_1)$, появляющееся в обеих таблицах, дает пару (k_1, k_2) ключей, переводящую P_1 в C_1. Множество общих значений этих таблиц содержит истинное значение $E^1_{k_1}(P_1)$ и случайное (зависящее от P_1) число других («ложных») значений.

Математическое ожидание числа таких ложных значений имеет порядок $O(2^{2n-m})$. Если пара (k_1, k_2) не эквивалентна истинному ключу (K_1, K_2), а P_2 выбирается независимо от P_1 и имеет равномерное распределение на множестве всех блоков из m битов, то применение пары ключей (k_1, k_2) к P_2 даст C_2 с вероятностью 2^{-m}. Таким образом, проверка на паре (P_2, C_2) уменьшает среднее число «ложных» пар (k_1, k_2) до $O(2^{2n-2m})$. Если $n > m$, то число «ложных» пар можно еще больше уменьшить, проверяя их на третьей паре (P_3, C_3), и т. д.

Для реализации описанного метода требуется память, достаточная для хранения таблиц, содержащих по 2^n двоичных строк по m битов каждая. В работе [2] предложена модификация этого метода, тоже основанная на задаче о днях рождения, позволяющая значительно уменьшить объем используемой памяти, не увеличивая существенно число операций.

В работе [3] предложен новый способ использования задачи о днях рождения, позволяющий находить решение системы

$$x_1 \oplus x_2 \oplus \ldots \oplus x_{2^k} = 0, \quad x_s \in L_s, \quad s = 1, \ldots, 2^k,$$

где $L_1, \ldots, L_{2^k}$ – заданные подмножества n-мерных двоичных векторов, за $O\left(2^{k+n/(k+1)}\right)$ операций.

В докладе предполагается обсудить эти новые и некоторые другие [4, 5] способы применения задачи о днях рождения в криптографическом анализе.

Литература

[1] Шнайер Б. Прикладная криптография. М.: Триумф, 2002.

[2] van Oorschot P. C., Wiener M. J. Improving implementable meet-in-the-middle attacks by order of magnitude. CRYPTO'96. Lect. Notes Comp. Sci., 1996, v. 1109, p. 229–236.

[3] Wagner D. A generalized birthday problem. CRYPTO'2002. Lect. Notes Comp. Sci., 2002, v. 2442, p. 288–303.

[4] Girault M., Cohen R., Campana M. A generalized birthday attack. EUROCRYPT'88. Lect. Notes Comp. Sci., 1988, v. 330, p. 129–158.

[5] Coppersmith D. Another birthday attack. CRYPTO'85. Lect. Notes Comp. Sci. 1986, v. 218, p. 14–17.