Московский государственный университет
им. М. В. Ломоносова
Институт проблем информационной безопасности МГУ
Аппарат Национального антитеррористического комитета
Академия криптографии Российской Федерации

Четвертая международная научная конференция по проблемам безопасности и противодействия терроризму

Московский государственный университет
им. М. В. Ломоносова, 30–31 октября 2008 г.

Том 2

Материалы Седьмой общероссийской научной конференции «Математика и безопасность информационных технологий» (МаБИТ-2008)

Москва
Издательство МЦНМО
2009

ББК 32.81В6
М34

Организация и проведение Седьмой Общероссийской научной конференции «Математика и безопасность информационных технологий» (МаБИТ-08) *были поддержаны грантом РФФИ* № 08-01-06116-*г.*

РФФИ

М34 **Материалы** Четвертой международной научной конференции по проблемам безопасности и противодействия терроризму. Московский государственный университет им. М. В. Ломоносова. 30—31 октября 2008 г. Том 2. Материалы Седьмой общероссийской научной конференции «Математика и безопасность информационных технологий» (МаБИТ-2008). — М.: МЦНМО, 2009. — 280 с.

ISBN 978-5-94057-503-0

ISBN 978-5-94057-501-6
ISBN 978-5-94057-503-0 (Том 2)

Содержание

Общая информация о Четвертой международной научной конференции по проблемам безопасности и противодействия терроризму

Мероприятия конференции:

- Семинар «Социально-философское обоснование методов противодействия религиозному экстремизму».
- Семинар-круглый стол «Роль средств массовой информации в профилактике терроризма».
- Семинар-круглый стол «Социально-психологические технологии профилактики терроризма».
- Семинар «Формирование идеологии антитеррористической: поиск оснований».
- Семинар-круглый стол «Мировая культура против идеологии терроризма».
- Международный круглый стол «Противодействие использованию сети Интернет в террористических целях».
- Первая всероссийская научно-практическая конференция «Формирование устойчивой антитеррористической позиции гражданского общества как основы профилактики терроризма».
- Седьмая общероссийская научная конференция «Математика и безопасность информационных технологий» (МаБИТ-2008), включающая секции по следующим тематическим направлениям:
  - — математические проблемы информационной безопасности;
  - — математическое и программное обеспечение информационной безопасности компьютерных систем.

Сопредседатели конференции:

- В. А. Садовничий — ректор МГУ имени М. В. Ломоносова;
- В. П. Шерстюк — помощник Секретаря Совета Безопасности РФ;
- С. М. Буравлев — заместитель Директора ФСБ РФ, президент Академии криптографии РФ;
- Е. П. Ильин — первый заместитель руководителя аппарата Национального антитеррористического комитета.

Оргкомитет конференции:

- В. В. Белокуров — сопредседатель Оргкомитета, проректор МГУ;
- Н. В. Семин — сопредседатель Оргкомитета, проректор МГУ;
- В. Н. Сачков — сопредседатель Оргкомитета, вице-президент Академии криптографии РФ;
- В. В. Ященко — сопредседатель Оргкомитета, зам. директора ИПИБ МГУ;
- А. А. Стрельцов (аппарат Совета Безопасности РФ);
- М. М. Глухов (Академия криптографии РФ);
- В. К. Левин (Академия криптографии РФ);
- В. И. Орлов (аппарат Национального антитеррористического комитета);
- В. Ю. Соколов (аппарат Национального антитеррористического комитета);
- В. В. Миронов (философский факультет МГУ);
- А. В. Сурин (факультет государственного управления МГУ);
- Ю. П. Зинченко (факультет психологии МГУ);
- Е. Л. Вартанова (факультет журналистики МГУ);
- В. Б. Алексеев (факультет ВМиК МГУ);
- В. А. Васенин (ИПИБ МГУ);
- Г. М. Кобельков (механико-математический факультет МГУ);
- И. Б. Котлобовский (экономический факультет МГУ);
- А. П. Лободанов (факультет искусств МГУ);
- О. А. Логачёв (ИПИБ МГУ);

- А. А. Сальников (ИПИБ МГУ);
- В. В. Соколов (ИПИБ МГУ);
- Д. И. Григорьев (ИПИБ МГУ);
- С. Г. Тер-Минасова (факультет иностранных языков и регионоведения МГУ);
- Е. Н. Мощелков (общественно-политический центр МГУ);
- Р. Перл (ОБСЕ);
- Р. Госенда (университет штата Нью-Йорк, США);
- Ш. Кросс (Центр им. Джорджа К. Маршалла);
- Г. Бехманн (университет г. Карлсруэ, Германия);
- Э. фон Штудниц (германско-российский форум);
- В. Марковский (ICANN — корпорация Интернета для специализированных адресов и номеров).

Секретариат конференции:

- Р. А. Шаряпов — отв. секретарь Оргкомитета (ИПИБ МГУ);
- Т. В. Крюкова — зам. отв. секретаря Оргкомитета;
- В. И. Солодовников (Академия криптографии РФ);
- А. В. Меркулов (аппарат Национального антитеррористического комитета);
- М. И. Анохин
- Т. А. Браташ
- О. В. Казарин
- Н. Н. Костина
- А. В. Соколова
- Н. В. Табаченко

Программа Седьмой общероссийской научной конференции «Математика и безопасность информационных технологий» (МаБИТ-2008)

Четверг, 30 октября 2008 г.

15.00–18.00. Секционное заседание «Математические проблемы информационной безопасности»

Место проведения: 2-й учебный корпус МГУ, аудитория П-8а.

Сопредседатели: О. А. Логачёв (зав. отделом ИПИБ МГУ, кандидат физико-математических наук, с. н. с.), В. Б. Алексеев (зав. кафедрой факультета ВМиК МГУ, доктор физико-математических наук, профессор).

А. Н. Алексейчук, Л. В. Ковальчук, Л. В. Скрыпник, А. С. Шевцов (Национальный технический университет Украины «Киевский политехнический институт»). Оценки практической стойкости блочного шифра «Калина» относительно разностного, линейного и билинейного методов криптоанализа.

М. А. Черепнёв (механико-математический факультет МГУ). Оценки времени работы нового алгоритма решения больших разреженных систем линейных уравнений над GF(2).

Г. Б. Маршалко (научное издательство «ТВП», г. Москва). О числе отрезков заданного ранга над кольцом вычетов.

И. В. Чижов (факультет ВМиК МГУ). Эквивалентные подпространства кода Рида — Маллера и множество открытых ключей криптосистемы Мак-Элиса — Сидельникова.

А. В. Халявин (механико-математический факультет МГУ). Неравенства для ортогональных массивов большой силы.

15.00–18.00. Секционное заседание «Математическое и программное обеспечение информационной безопасности компьютерных систем»

Место проведения: актовый зал НИИ механики МГУ.

Сопредседатели: В. К. Левин (академик РАН), В. А. Васенин (зав. отделом ИПИБ МГУ, доктор физико-математических наук, профессор).

А. В. Галатенко (механико-математический факультет МГУ, ИПИБ МГУ). О восстановлении разбиения множества состояний безопасности.

П. Д. Зегжда, М. О. Калинин, Д. А. Москвин (ГОУ СПбГПУ). Анализ способов управления безопасностью информационных систем с помощью методов многокритериальной оптимизации и аппарата графов.

П. Н. Девянин (ИКСИ). Применение базовой ролевой ДП-модели для анализа условий передачи прав доступа.

К. А. Шапченко, О. О. Андреев (механико-математический факультет МГУ, ИПИБ МГУ). Подход к управлению настройками механизмов безопасности в дистрибутивах ОС Linux.

В. А. Пономарев, О. Ю. Богоявленская, Ю. А. Богоявленский (Петрозаводский государственный университет). Конфигурируемая модульная система мониторинга поведения транспортного протокола на уровне ядра операционной системы.

С. А. Афонин (НИИ механики МГУ). О криптографии с открытым ключом на основе задачи разложения языков.

Пятница, 31 октября 2008 г.

10.00–17.30. Секционное заседание «Математические проблемы информационной безопасности»

Место проведения: 2-й учебный корпус МГУ, аудитория П-8а.

Сопредседатели: О. А. Логачёв (зав. отделом ИПИБ МГУ, кандидат физико-математических наук, с. н. с.), М. М. Глухов (ученый секретарь отделения Академии криптографии РФ, доктор физико-математических наук, профессор).

С. П. Горшков, А. В. Тарасов (Академия криптографии РФ). Теоретико-сложностной подход к оценке сложности решения систем булевых уравнений.

С. Н. Селезнёва (факультет ВМиК МГУ). Линейный алгоритм, определяющий по вектору значений булевой функции, задается ли она полиномом фиксированной степени.

Б. А. Погорелов (Академия криптографии РФ), М. А. Пудовкина (МИФИ). О метриках, изометричных относительно группы сдвигов.

С. В. Смышляев (факультет ВМиК МГУ). О некоторых свойствах совершенно уравновешенных булевых функций.

Г. А. Карпунин, Нгуен Т. Х. (факультет ВМиК МГУ). Оптимальность выбора функции XOR в одной модели дифференциального криптоанализа хэш-функций семейства MD*x*.

В. Е. Федюкович (GlobalLogic Ukraine, г. Киев). Протокол аргумента для цикла Гамильтона.

С. С. Коновалова, С. С. Титов (Уральский государственный университет путей сообщения, г. Екатеринбург). Комбинаторно-геометрический метод исследования взаимосвязей между шифрами.

14.00–15.00. Обед

Ю. С. Харин (Белорусский государственный университет, г. Минск). Дискретные временные ряды и их использование в задачах защиты информации.

Е. К. Алексеев (факультет ВМиК МГУ). О некоторых свойствах линейных кодов, образующих носители корреляционно-иммунных булевых функций.

Г. А. Карпунин, А. А. Мартыненко (факультет ВМиК МГУ). Стойкость криптосистемы МакЭлиса на основе недвоичных кодов Гоппы.

О. А. Логачёв (ИПИБ МГУ). Об использовании аффинных нормальных форм булевых функций для определения ключей фильтрующих генераторов.

10.00–17.30. Секционное заседание «Математическое и программное обеспечение информационной безопасности компьютерных систем»

Место проведения: актовый зал НИИ механики МГУ.

Сопредседатели: В. А. Васенин (зав. отделом ИПИБ МГУ, доктор физико-математических наук, профессор), Г. М. Кобельков (зав. кафедрой механико-математического факультета МГУ, доктор физико-математических наук, профессор).

В. А. Галатенко, К. А. Костюхин, А. С. Малиновский, Н. В. Шмырёв (НИИСИ РАН). Программирование, ориентированное на мониторинг, как элемент контролируемого выполнения аппаратно-программных комплексов.

А. В. Львова (ФГУП ВНИИПВТИ). Модель управления рисками информационной безопасности на основе знания угроз.

С. С. Корт, Е. А. Рудина (ГОУ СПбГПУ). Архитектура ядра системы мониторинга и защиты от вторжений.

11.30–11.45. Перерыв

О. Д. Соколова, А. Н. Юргенсон (ИВМ и МГ СО РАН). Задача оптимальной расстановки систем мониторинга потоков.

В. А. Десницкий, И. В. Котенко (СПИИРАН). Проектирование и анализ протокола удаленного доверия.

А. И. Тупицын (ФГУП «НИИ Квант»). Автоматизация процесса мониторинга информационной безопасности компьютерных систем на основе политик безопасности.

Д. В. Комашинский, И. В. Котенко (СПИИРАН). Исследование проактивных механизмов обнаружения вредоносного программного обеспечения на базе методов Data Mining.

14.00–15.00. Обед

П. Д. Зегжда, Е. А. Рудина (ГОУ СПбГПУ). Формальное представление сетевого протокола.

А. А. Чечулин, И. В. Котенко (СПИИРАН). Защита от сетевых атак методами фильтрации и нормализации протоколов транспортного и сетевого уровня стека TCP/IP.

А. А. Хураскин, М. С. Дзыба, А. А. Коршунов (механико-математический факультет МГУ, НИИ механики МГУ). Определение топологии компьютерных сетей на физическом уровне.

В. Г. Проскурин (ИКСИ). Антивирусная защита операционных систем штатными средствами.

В. Б. Савкин (НИИ механики МГУ). К вопросу имитационного моделирования механизмов разделения коммуникационных ресурсов компьютерных сетей.

Часть I

СЕКЦИЯ «МАТЕМАТИЧЕСКИЕ ПРОБЛЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ»

Оценки практической стойкости блочного шифра «Калина» относительно разностного, линейного и билинейного методов криптоанализа

А. Н. Алексейчук, Л. В. Ковальчук,
Л. В. Скрыпник, А. С. Шевцов

1. Блочный шифр (БШ) «Калина» [1] является одним из кандидатов на Национальный стандарт шифрования Украины. Существенной особенностью этого шифра, выделяющей его среди современных БШ, построенных в более традиционном стиле, является использование различных арифметических операций в соседних раундах шифрования. Как и в случае с алгоритмом шифрования ГОСТ 28147-89, подобное «смешивание» операций приводит к определенным затруднениям при оценке стойкости шифра относительно статистических методов криптоанализа (разностного, линейного и др. [2]). Аккуратное обоснование оценок практической стойкости БШ «Калина» относительно указанных методов требует применения более «тонкого» математического аппарата [3, 4], позволяющего преодолевать аналитические трудности в тех случаях, когда традиционные математические методы не приводят к успеху.

В докладе представлены результаты исследования семейства блочных шифров, построенных по схеме шифра «Калина» с длиной блока 128 бит («Калина-128»). Получены верхние оценки параметров, характеризующих практическую стойкость рассматриваемых БШ относительно разностного, линейного и билинейного методов криптоанализа (в последнем случае оцениваются числовые характеристики раундовых функций БШ). Применительно к шифру «Калина-128», численные значения верхних оценок средних вероятностей дифференциальных и линейных характеристик составляют 2^{-230} и 2^{-212} соответственно, что свидетельствует о практической стойкости этого шифра к классическим разностным и линейным атакам. Отметим также, что результаты, полученные для БШ «Калина-128», распространяются и на другие версии этого шифра (с длинами блоков 256 или 512 бит).

2. Пусть t, p', r' — натуральные числа. Обозначим $p = 4p'$, $r = 2r' + 1$, $n = pt$ и рассмотрим r-раундовый БШ $\mathfrak{Z}$ с множеством открытых (шифрованных) сообщений $V_n = \{0, 1\}^n$, множеством раундовых ключей $K = V_n$ и семейством шифрующих преобразований

$$F_{(k_1,\ldots,k_r)} = f_{r,k_r} \circ \ldots \circ f_{1,k_1},\ (k_1, \ldots, k_r) \in K^r, \tag{1}$$

где для любых $x \in V_n$, $k \in K$, $i \in \overline{1, r}$

$$f_{i,k}(x) = \begin{cases} \varphi(x \oplus k), & \text{если } i \equiv 1 \pmod 2,\ i < r; \\ \varphi\left(x \mathbin{\overset{\circ}{+}} k\right), & \text{если } i \equiv 0 \pmod 2; \\ s(x \oplus k), & \text{если } i = r. \end{cases} \tag{2}$$

В формулах (2) символы $\oplus$ и $\overset{\circ}{+}$ обозначают соответственно операцию покоординатного булевого сложения двоичных векторов длины n и алгебраическую операцию вида

$$x \mathbin{\overset{\circ}{+}} k = \left(x^{(1)} + k^{(1)}, \ldots, x^{(4)} + k^{(4)}\right), \tag{3}$$

где $x = \left(x^{(1)}, \ldots, x^{(4)}\right)$, $k = \left(k^{(1)}, \ldots, k^{(4)}\right)$, $x^{(\nu)}$, $k^{(\nu)} \in V_{tp'}$, $\nu \in \overline{1, 4}$, а $+$ есть символ операции сложения по модулю $2^{tp'}$ на множестве $V_{tp'}$. Далее, подстановки φ и s определяются по формулам

$$\varphi(x) = s(x)M, \quad x \in V_n, \tag{4}$$

$$s(x) = (s_0(x_0), \ldots, s_{p-1}(x_{p-1})), \quad x = (x_0, \ldots, x_{p-1}), \tag{5}$$

где $x_j \in V_t$, s_j — подстановка на множестве V_t (s-блок), $j \in \overline{0, p-1}$, а M есть обратимая $(p \times p)$-матрица над полем $\mathrm{GF}(2^t)$, и умножение $s(x)$ на M в выражении (4) осуществляется над этим полем (при естественном отождествлении двоичных векторов $s_j(x_j)$, $j \in \overline{0, p-1}$, с его элементами). Примером шифра, удовлетворяющего описанным условиям, является «Калина-128» [1]. В этом случае $t = 8$, $p' = 4$, $r' = 5$ ($n = 128$, $r = 11$), а матрица M имеет следующий вид:

$$M = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & I & 0 \\ 0 & I & 0 & 0 \end{pmatrix} \mathrm{diag}(D, D), \tag{6}$$

где I и 0 — соответственно единичная и нулевая матрицы порядка 4 над полем $\mathrm{GF}\left(2^8\right)$, D — МДР-матрица 8-го порядка над этим полем.

Напомним, что средняя вероятность дифференциальной характеристики $\Omega = (\omega_0, \omega_1, \ldots, \omega_r) \in (V_n \setminus \{0\})^{r+1}$ БШ $\mathfrak{I}$ определяется по формуле [5]

$$\mathrm{EDP}(\Omega) = |K|^{-r} \sum_{(k_1,\ldots,k_r)\in K^r} \mathrm{DP}^{(k_1,\ldots,k_r)}(\Omega), \tag{7}$$

где

$$\mathrm{DP}^{(k_1,\ldots,k_r)}(\Omega) = \mathsf{P}(X_r \oplus X'_r = \omega_r, \ldots, X_1 \oplus X'_1 = \omega_1 \,|\, X \oplus X' = \omega_0),$$

X, X' — независимые случайные равновероятные двоичные векторы длины n, $X_i = (f_{i,k_i} \circ \ldots \circ f_{1,k_1})(X)$, $X'_i = (f_{i,k_i} \circ \ldots \circ f_{1,k_1})(X')$, $i \in \overline{1, r}$. Средней вероятностью линейной характеристики $\Omega = (\omega_0, \omega_1, \ldots, \omega_r)$ БШ $\mathfrak{I}$ называется величина

$$\mathrm{ELP}(\Omega) = \prod_{i=1}^{r} \left(2^{-n} \sum_{k\in V_n} \left(2^{-n} \sum_{x\in V_n} (-1)^{\omega_{i-1} x \oplus \omega_i f_{i,k}(x)} \right)^2 \right). \tag{8}$$

Параметры (7), (8) являются стандартными показателями практической стойкости блочных шифров относительно разностного и линейного криптоанализа соответственно (см., например, [2]). Необходимым условием практической стойкости БШ $\mathfrak{I}$ относительно билинейных атак [6] является отсутствие так называемых высоковероятных билинейных аппроксимаций отображения (5), то есть таких уравновешенных функций $B_{A,\alpha,\beta}(x, y) = xAy \oplus \alpha x \oplus \beta y$, $(x, y) \in V_n \times V_n$, где A — ненулевая булева матрица порядка n, $\alpha, \beta \in V_n$, для которых значение параметра

$$l_*^{(s)}(A, \alpha, \beta) = 2^{-n} \sum_{k\in V_n} \left(2^{-n} \sum_{x\in V_n} (-1)^{B_{A,\alpha,\beta}(x, s(s*k))} \right)^2, \quad * \in \left\{ \oplus, \overset{\circ}{+} \right\}, \tag{9}$$

достаточно близко к 1.

Таким образом, для оценки практической стойкости БШ $\mathfrak{I}$ относительно перечисленных методов криптоанализа требуется получить аналитические верхние границы параметров (7), (8) и (9).

3. Введем ряд дополнительных обозначений. Для любых u, $v \in V_t$ обозначим $u + v$ сумму по модулю 2^t двоичных целых чисел, соответствующих векторам u и v; символом $\nu(u, v)$ обозначим бит переноса в t-й разряд при сложении чисел u и v в кольце $\mathbf{Z}$.

Для любого $j \in \overline{0,\, p-1}$ положим

$$d_*^{(s_j)} = \max_{\alpha,\beta \in V_t \setminus \{0\}} 2^{-t} \sum_{k \in V_t} \delta(s_j(k * \alpha) \oplus s_j(k),\, \beta), \quad * \in \{\oplus,\, +\},$$

$$l_\oplus^{(s_j)} = \max_{\alpha,\beta \in V_t \setminus \{0\}} 2^{-t} \sum_{k \in V_t} \left(2^{-t} \sum_{x \in V_t} (-1)^{\alpha x \oplus \beta s_j(x \oplus k)} \right)^2,$$

$$\Lambda^{(s_j)} = \max_{\alpha,\beta \in V_t \setminus \{0\}} 2^{-t} \sum_{k \in V_t} \left(2^{-t} \sum_{a \in \{0,1\}} \left| \sum_{\substack{x \in V_t: \\ \nu(x,k)=a}} (-1)^{\alpha x \oplus \beta s_j(x+k)} \right| \right)^2,$$

где $\delta(u,\, v)$ — символ Кронекера. Положим также

$$\Delta_\oplus = \max_{j \in \overline{0,p-1}} d_\oplus^{(s_j)}, \quad \Delta_+ = \max_{j \in \overline{0,p-1}} d_+^{(s_j)},$$

$$\Lambda_\oplus = \max_{j \in \overline{0,p-1}} l_\oplus^{(s_j)}, \quad \Lambda_+ = \max_{j \in \overline{0,p-1}} \Lambda^{(s_j)},$$

$$\Delta = \max\{\Delta_\oplus,\, \Delta_+\}, \quad \Lambda = \max\{\Lambda_\oplus,\, \Lambda_+\}. \tag{10}$$

Напомним, что вес вектора $x = (x_0,\, \ldots,\, x_{p-1})$, где $x_j \in \mathrm{GF}(2^t)$, $j \in \overline{0,\, p-1}$, определяется по формуле $\mathrm{wt}(x) = \#\left\{ j \in \overline{0,\, p-1} : x_j \neq 0 \right\}$, а индекс ветвления (branch number) обратимой матрицы M — по формуле [7]

$$B_M = \min_{x \in \mathrm{GF}(2^t)^p \setminus \{0\}} \mathrm{wt}(x) + \mathrm{wt}(xM^{-1}). \tag{11}$$

Следующая теорема устанавливает верхние оценки параметров (7), (8).

Теорема 1. *Пусть* $\mathfrak{I}$ *— r-раундовый БШ, описываемый соотношениями* (1)–(6), $r = 2r' + 1$. *Тогда справедливы неравенства*

$$\mathrm{EDP}(\Omega) \leqslant \Delta^{r' B_M + 1}, \quad \mathrm{ELP}(\Omega) \leqslant \Lambda^{r' B_M + 1}, \tag{12}$$

где Δ *и* Λ *определяются по формулам* (10), *а* B_M *— по формуле* (11).

Отметим, что в доказательстве теоремы 1 существенно используются результаты, изложенные в [3, 4], а также тот факт, что при выполнении условий (1)–(6) $\mathfrak{I}$ является обобщенным марковским шифром [4] относительно каждой из операций $\oplus$, $\overset{\circ}{+}$. Численные расчеты значений (10) для шифра «Калина-128» показывают, что $\Delta = 0,031250$, $\Lambda = 0,0421770$. Отсюда на основании (12) (при $r' = 5$, $B_M = 9$ [1]) вытекают оценки $\mathrm{EDP}(\Omega) \leqslant 2^{-230}$, $\mathrm{ELP}(\Omega) \leqslant 2^{-212}$, свидетельствующие о том, что рассматриваемый БШ является практически стойким относительно разностного и линейного методов криптоанализа.

4. Приведем оценки параметра (9), справедливые при некоторых ограничениях на матрицу A. Запишем эту матрицу в блочном виде: $A = (A_{ij})_{i,j\in\overline{0,p-1}}$, где A_{ij} — матрица порядка t; аналогично представим векторы α и β: $\alpha = (\alpha_0, \ldots, \alpha_{p-1})$, $\beta = (\beta_0, \ldots, \beta_{p-1})$ где $\alpha, \beta \in V_t$, $j \in \overline{0, p-1}$. Для любого $i \in \overline{0, p-1}$ обозначим $C_i(A)$ ($S_i(A)$) подпространство векторного пространства V_t, порожденное столбцами матриц A_{ij}(строками матриц A_{ji}) по всем $j \neq i$. Введем также следующие обозначения:

$$l_*^{(s_j)}(B, u, v) = 2^{-t} \sum_{k\in V_t} \left(2^{-t} \sum_{x\in V_t} (-1)^{xBs_j(x*k)\oplus ux\oplus vs_j(x*k)} \right)^2, \quad * \in \{\oplus, +\},$$

$$\Lambda^{(s_j)}(B, u, v) = 2^{-t} \sum_{k\in V_t} \left(2^{-t} \sum_{a\in\{0,1\}} \left| \sum_{\substack{x\in V_t:\\ \nu(x,k)=a}} (-1)^{xBs_j(x+k)\oplus ux\oplus vs_j(x+k)} \right| \right)^2,$$

где B — произвольная булева матрица порядка t, $u, v \in V_t$, $j \in \overline{0, p-1}$.

Теорема 2. *Пусть матрица* $A = (A_{ij})_{i,j\in\overline{0,p-1}}$ *удовлетворяет условию*

$$\exists\, i \in \overline{0, p-1} : (\forall\, j < i : A_{ij} = 0) \vee (\forall\, j < i : A_{ji} = 0). \tag{13}$$

Тогда для любых $\alpha, \beta \in V_n$ *справедливы неравенства*

$$l_{\oplus}^{(s)}(A, \alpha, \beta) \leqslant \max_{u\in C_i(A), v\in S_i(A)} l_{\oplus}^{(s_i)}(A_{ii}, \alpha_i \oplus u, \beta_i \oplus v), \tag{14}$$

$$l_{\substack{\circ\\+}}^{(s)}(A, \alpha, \beta) \leqslant \max_{u\in C_i(A), v\in S_i(A)} \Lambda^{(s_i)}(A_{ii}, \alpha_i \oplus u, \beta_i \oplus v). \tag{15}$$

Кроме того, если $A_{ij} = 0$ *для любых* $i \neq j$, *то*

$$l_{\substack{\circ\\+}}^{(s)}(A, \alpha, \beta) \leqslant l_{+}^{(s_0)}(A_{00}, \alpha_0, \beta_0) \prod_{i=1}^{p-1} \Lambda^{(s_i)}(A_{ii}, \alpha_i, \beta_i);$$

последняя оценка достигается, если $A_{ii} = 0$, $\alpha_i = \beta_i = 0$ *для всех* $i \in \overline{1, p-1}$.

Вычисление значений параметров $\max_{u,v\in V_8} l_{\oplus}^{(s_i)}(B, u, v)$ и $\max_{u,v\in V_8} \Lambda^{(s_i)}(B, u, v)$ для s-блоков s_i шифра «Калина-128» и нескольких десятков случайно сгенерированных ненулевых (8×8)-матриц B показывает, что эти значения не превосходят $0,004$. Отсюда на основании (14), (15) вытекают аналогичные оценки параметра (9), справедливые для любой матрицы A, удовлетворяющей условию (13) и содержащей любую из сгенерированных подматриц на главной диагонали.

В целом, результаты проведенных исследований свидетельствуют об отсутствии ярко выраженных слабостей шифра «Калина» относительно билинейного метода криптоанализа.

Литература

[1] *Горбенко І. Д., Долгов В. І., Олійников Р. В., Руженцев В. І., Михайленко М. С., Горбенко Ю. І., Тоцький О. С., Казьміна С. В.* Перспективний блоковий симетричний шифр «Калина» — основні положення та специфікації // Прикладная радиоэлектроника, 2007, т. 6, № 2, с. 195–208.

[2] *Wagner D.* Towards a unifying view of block cipher cryptanalysis // Proceedings of FSE'04, Springer-Verlag, 2004, p. 116–135.

[3] *Alekseychuk A. N., Kovalchuk L. V.* Upper bounds of maximum values of average differential and linear characteristic probabilities of Feistel cipher with adder modulo 2^m // Theory of Stohastic Processes, 2006, v. 12(28), № 1, 2, p. 20–32.

[4] *Ковальчук Л. В.* Обобщенные марковские шифры: построение оценки практической стойкости относительно дифференциального криптоанализа // Математика и безопасность информационных технологий. Материалы конференции в МГУ 25–27 октября 2006 г., М.: МЦНМО, 2007, с. 295–299.

[5] *Vaudenay S.* On the security of CS-cipher // Proceedings of FSE'99, Springer-Verlag, 1999, p. 260–274.

[6] *Courtois N. T.* Feistel schemes and bi-linear cryptanalysis // Proceedings of CRYPTO'04, Springer-Verlag, 2004, p. 23–40.

[7] *Daemen J.* Cipher and hash function design strategies based on linear and differential cryptanalysis. Doctoral dissertation, 1995.

Алгоритмы построения матричных приближений Паде

М. А. Черепнёв

Аннотация

В докладе описываются два новых алгоритма построения матричных приближений Паде для рядов по отрицательным степеням переменной.

При решении задачи факторизации целых чисел возникает необходимость решать систему линейных однородных разреженных уравнений над полем $\mathbb{F} = \mathrm{GF}(2)$. Ранее (см. [1]) в докладах автора было показано как эту задачу свести к построению матричных приближений формальных рядов с матричными коэффициентами по отрицательным степеням переменной. Основная часть алгоритма — это процедура построения следующего приближения при помощи предыдущих. Для получения решения исходной линейной системы необходимо построить приближения с невырожденными старшими коэффициентами. Рассмотрим несколько последовательных приближений Паде:

$$\alpha(\lambda)Q^{(s)}(\lambda) - P^{(s)}(\lambda) = \sum_{i=s+1}^{\infty} \sigma_i \lambda^{-i},$$

где λ — формальная переменная, греческими буквами обозначены матрицы из $\mathbb{F}(n \times n)$, большими латинскими — матричные полиномы из $\mathbb{F}(n \times n)[\lambda]$, степени которых не превосходят их номеров. Считаем, что коэффициенты разложений справа — независимые случайные матрицы.

Пусть $Q^{(-1)} = 0_n$, $P^{(-1)} = I_n$, $Q^{(0)} = I_n$, $P^{(0)} = \alpha_0$, где I_n, 0_n — соответственно единичная и нулевая матрицы. Был предложен алгоритм построения очередного приближения Паде при помощи модифицированных предыдущих. Было доказано, что приближение с номером $r+1$ может быть построено при помощи приближений с номерами r, $r-1$, …, $r-t$, если некоторые величины, характеризующие вырожденность старших коэффициентов предыдущих приближений $\xi_1, \xi_2, \ldots, \xi_t$ удовлетворяют следующим неравенствам:

$$\xi_1 \leqslant t-1,\ \xi_2 \leqslant t-1,\ \xi_3 \leqslant t-2,\ \ldots,\ \xi_t \leqslant 1.$$

Обозначим

$$\theta = \min\{t \geqslant 1 : \xi_1 \leqslant t-1,\ \xi_k \leqslant t-k+1,\ k = 2, \ldots, t\}.$$

Идея следующей теоремы предложена А. М. Зубковым.

Теорема. *В предположении случайности и независимости матричных коэффициентов рассматриваемых приближений, матожидание числа последовательных приближений необходимых для построения очередного при помощи рассматриваемого алгоритма удовлетворяет следующему неравенству*:

$$M(\theta) \leqslant 4,82.$$

Доказательство использует несколько известных оценок распределения коранга случайных квадратных матриц [2]:

$$\begin{aligned}
\rho &= \mathsf{P}(\operatorname{corank} M \geqslant 2) \leqslant \frac{1}{4},\\
\sigma &= \mathsf{P}(\operatorname{corank} M > 0) \leqslant 0,72,\\
\delta &= \mathsf{P}(\operatorname{corank} M = 0) \leqslant \frac{1}{2},\\
&\quad \mathsf{P}(\operatorname{corank} M = 1) \leqslant 0,6.
\end{aligned}$$

В докладе предложен ещё один алгоритм построения таких приближений.

Литература

[1] *Черепнев М. А.* Блочный алгоритм типа Ланцоша решения разреженных систем линейных уравнений // Дискрет. матем. 2008. Т. 20, № 1. С. 145–150.

[2] *Алексейчук А. Н.* Неасимптотические оценки распределения вероятностей ранга случайной матрицы над конечным полем // Дискрет. матем. 2007. Т. 19, № 2. С. 85–93.

О числе отрезков заданного ранга над кольцом вычетов

Г. Б. Маршалко

В работе [1] получено точное значение числа последовательностей длины n (n-последовательностей) заданного ранга (под рангом понимается степень минимального многочлена отрезка) над конечным полем. В работе представленной данным сообщением исследуются вопросы связанные с вычислением значения числа n-последовательностей заданного ранга над кольцом вычетов $\mathbb{Z}_N$, $N = p_1^{m_1} \dots p_t^{m_t}$, $p_1, \dots, p_t$ — различные простые числа.

В силу изоморфизма кольца $\mathbb{Z}_N$ внешней прямой сумме примарных колец вычетов $\mathbb{Z}_{p_i^{m_i}}$, $i = \overline{1, t}$, изучение ранга n-последовательности $u^n \in \mathbb{Z}_N$ естественным образом сводится к изучению ранга ее компонент над примарным кольцом $\mathbb{Z}_{p^m}$.

Для дальнейшего изложения нам потребуется краткое описание алгоритма Берлекемпа—Месси нахождения минимального многочлена последовательности. Мы будем использовать разработанную В. Л. Куракиным версию алгоритма для конечных колец с единицей, которую можно найти, например, в [2].

Определение 1. Элементы $a, b \in \mathbb{Z}_{p^m}$ называются ассоциированными, если существует обратимый элемент $v \in \mathbb{Z}_{p^m}^*$, такой что $a = vb$.

Отношение ассоциированности, заданное в соответствии с определением 1, разбивает множество элементов кольца на непересекающиеся классы $K_0, K_1, \dots, K_{m-1}$. При этом будем считать, что классу K_0 принадлежат обратимые элементы кольца, а остальным классам — делители нуля. Обозначим $w_i = |K_i|$, $i = \overline{0, m-1}$ — мощность соответствующего класса, а число элементов в кольце через $R = p^m$. Будем также считать, что класс K_i, $i = \overline{1, m-1}$, содержит элементы с индексом нильпотентности равным $m + 1 - i$.

Алгоритм Берлекемпа—Месси заполняет для каждого $i = \overline{0, m-1}$ таблицу, соответствующую классу K_i (см. табл. 1, где $a_i \in K_i$). На s-й строке этой таблицы находятся отрезок и многочлен, с помощью которого данный отрезок получается из первого отрезка. Кроме этого заполняется таб-

Работа выполнена при поддержке гранта Президента РФ НШ 4.2008.10.

лица, содержащая идеалы $I_s(k)$, $s = 0, 1, \ldots, k = \overline{0, n-s-1}$ (см. табл. 2). Идеал $I_s(k)$ порожден элементами v такими, что на шаге $l < s$ появился отрезок, в начале которого было k нулей, а на $k+1$ месте находился элемент v.

Таблица 1

Номер шага s	Отрезок	Многочлен
0	$u^n(0, i) = a_i u^n$	$f_{0i}(x) = a_i$
1	$u^{n-1}(1, i)$	$f_{1i}(x) = a_i x + \ldots$
…		
s	$u^{n-s}(s, i)$	$f_{si}(x) = a_i x^s + \ldots$

Таблица 2

s	$I_s(0)$	$I_s(1)$	…	$I_s(n-2)$	$I_s(n-1)$
0	$I_0(0)$	$I_0(1)$		$I_0(n-2)$	$I_0(n-1)$
1	$I_1(0)$	$I_1(1)$		$I_1(n-2)$	
2	$I_2(0)$	$I_2(1)$			

Алгоритм заключается в следующем. На s-м шаге отрезок в текущей таблице сдвигается на один элемент влево (умножается на x), и, если первый ненулевой элемент принадлежит соответствующему идеалу, то данный элемент может быть обнулен с помощью всех отрезков полученных на предыдущих этапах алгоритма во всех таблицах. После обработки всех таблиц подправляется таблица идеалов: в соответствующие идеалы добавляются первые ненулевые элементы получившихся отрезков. Алгоритм заканчивает работу, когда в первой таблице появляется нулевой отрезок.

Всюду далее считаем, что алгоритм закончил работу на шаге $r \in \overline{0, n}$. Справедливы следующие леммы.

Лемма 1. *Если построенный в результате работы алгоритма Берлекемпа—Месси идеал $I_{r-1}(k)$ порожден элементом a, то идеал $I_{r-1}(k-1)$ также порожден этим элементом.*

Лемма 2. $I_{r-1}(r) = 0$.

Из последней леммы следует, что если ранг последовательности u^n равен r, то ненулевыми являются идеалы $I_{r-1}(0), \ldots, I_{r-1}(r-1)$.

Определение 2. i-м подрангом последовательности u^n будем называть величину

$$r_i = \max_j \{a_i \in I_{r-1}(j),\ a_i \in K_i\}, \quad i = \overline{0, m-1}.$$

Следствие 1. $r_0 \leqslant r_1 \leqslant \ldots \leqslant r_{m-1} = r$.

Далее для простоты изложения будем записывать $(r_0, r_1, \ldots, r_{m-1})(u^n)$ в случае, если последовательность u^n имеет подранги $r_0, r_1, \ldots, r_{m-1}$.

Заметим, что если мы рассмотрим все $(n+1)$-последовательности $u^n a$, $a \in \mathbb{Z}_{p^m}$, то в результате работы алгоритма Берлекемпа—Месси на r-том шаге для последовательнсотей такого вида в первой таблице получатся $p^m(n+1-r)$-последовательностей вида $0, \ldots, 0, b$, $b \in \mathbb{Z}_{p^m}$.

Утверждение 1. *Пусть n-последовательность u^n имеет подранги $(r_0, r_1, \ldots, r_{m-1})(u^n)$. Тогда*

1) *если $r_0 \leqslant r_1 \leqslant \ldots \leqslant r_{m-1} \leqslant [(n+1)/2]$, то $(n+1)$-последовательность $u^n a$ имеет подранги*
   - $(r_0, r_1, \ldots, r_{m-1})(u^n)$ *при* $b = 0$;
   - $(r_0, r_1, \ldots r_{i-1}, n+1-r_i, \ldots, n+1-r_i)(u^n)$ *при* $b \in K_i$, $i = \overline{0, m-1}$;
2) *если* $r_0 \leqslant \ldots \leqslant r_{i-1} \leqslant [(n+1)/2] < r_i \leqslant \ldots \leqslant r_{m-1}$, *то* $(n+1)$-*последовательность $u^n a$ имеет подранги*
   - $(r_0, r_1, \ldots, r_{m-1})(u^n)$, $b = 0$, $b \in K_s$, $s = \overline{i, m-1}$ *или* $b \in K_s$, $s = \overline{0, i-1}$ *и* $r_s \geqslant n+1-r_{m-1}$;
   - $(r_0, r_1, \ldots r_{s-1}, n+1-r_s, \ldots, n+1-r_s)(u^n)$, $b \in K_s$, $s = \overline{0, i-1}$ *и* $r_s < n+1-r_{m-1}$;
3) *если* $[(n+1)/2] < r_0 \leqslant r_2 \leqslant \ldots \leqslant r_{m-1}$, *то* $(n+1)$-*последовательность $u^n a$ имеет подранги* $(r_0, r_1, \ldots, r_{m-1})(u^n)$.

Данное утверждение описывает определяющие соотношения для ранга n-последовательности при увеличении ее длины и позволяет получить выражения для точного числа последовательностей заданного ранга. Покажем на примере кольца $\mathbb{Z}_{p^2}$ методику расчета указанных величин. Обозначим через $N_n(r_0, r_1)$ число n-последовательностей длины имеющих подранги r_0, r_1. Положим $N_n(r_0, n+1) = 0$.

Следствие 2. *Если n четно, то $N_n(r_0, r_1) = 0$, $r_1 > n/2$, $n - r_1 < r_0 < r_1$. Если же n нечетно, то $N_n(r_0, r_1) = 0$, $r_1 > [n/2] + 1$, $n - r_1 < r_0 < r_1$.*

Следствие 3. *При $n > 1$ число $(n+1)$-последовательностей, имеющих подранги r_0, r_1, удовлетворяет следующим равенствам.*

1. *Пусть $n+1$ нечетно. Тогда*
   - *если $r_0 \leqslant r_1 \leqslant [(n+1)/2]$, то $N_{n+1}(r_0, r_1) = N_n(r_0, r_1)$;*

- *если* $r_0 \leqslant [(n+1)/2] < r_1$, *то*

$$N_{n+1}(r_0, r_1) = \begin{cases} (w_1+1)N_n(r_0, r_1) \\ \qquad + w_1 N_n(r_0, n+1-r_1), & r_0 \leqslant n+1-r_1; \\ 0, & r_0 > n+1-r_1; \end{cases}$$

- *если* $[(n+1)/2] < r_0 < r_1$, *то* $N_{n+1}(r_0, r_1) = 0$;
- *если* $[(n+1)/2] < r_0 = r_1 = r$, *то*

$$N_{n+1}(r, r) = (w_0 + w_1 + 1)N_n(r, r) + w_0 \sum_{k=n+1-r}^{r} N_n(n+1-r, k).$$

2. *Пусть* $n+1$ *четно. Тогда*

- *если* $r_0 \leqslant r_1 \leqslant (n+1)/2$, *то*

$$N_{n+1}(r_0, r_1) = N_n(r_0, r_1) + \begin{cases} w_1 N_n(r_0, r_1), & r_0 < r_1 = (n+1)/2; \\ (w_0 + w_1)N_n(r_0, r_1), & r_0 = r_1 = (n+1)/2; \end{cases}$$

- *если* $r_0 \leqslant (n+1)/2 < r_1$, *то*

$$N_{n+1}(r_0, r_1) = \begin{cases} (w_1+1)N_n(r_0, r_1) \\ \qquad + w_1 N_n(r_0, n+1-r_1), & r_0 \leqslant n+1-r_1; \\ 0, & r_0 > n+1-r_1; \end{cases}$$

- *если* $(n+1)/2 < r_0 < r_1$, *то* $N_{n+1}(r_0, r_1) = 0$;
- *если* $(n+1)/2 < r_0 = r_1 = r$, *то*

$$N_{n+1}(r, r) = (w_0 + w_1 + 1)N_n(r, r) + w_0 \sum_{k=n+1-r}^{r} N_n(n+1-r, k).$$

Здесь w_0 *и* w_1 *— определенные ранее мощности классов ассоциированных элементов.*

Лемма 3.

$$N_n(r, r) = \begin{cases} 1, & r = 0; \\ R^{2r-1} w_0, & n \geqslant 2r. \end{cases}$$

Следствие 4.

$$N_n(r_0,\, r_1) = \begin{cases} N_{2r_0}(r_0,\, r_0), & r_1 = r_0; \\ w_1(w_1+1)^{2r_1-2r_0-1} N_{2r_0}(r_0,\, r_0), & r_0 < r_1 \leqslant [n/2]; \\ w_1(w_1+1)^{2n-2r_1-2r_0} N_{2r_0}(r_0,\, r_0), & r_1 > [n/2],\ r_0 \leqslant n - r_1; \\ 0, & r_1 > [n/2], \\ & \quad n - r_1 < r_0 < r_1. \end{cases}$$

Следствие 5. $N_n(r,\, r) = R^{2n-2r} w_0 (w_1+1)^{2r-n-1}$, $r \geqslant n/2$.

Утверждение 2. *Число $N_n(r)$ n-последовательностей ранга r над кольцом $\mathbb{Z}_{p^2}$ равно* 1, *если* $r = 0$,

$$\frac{R^{2r+1} - R^{2r-1}(w_1+1) + w_1(w_1+1)^{2r}}{R + w_1 + 1},$$

если $r \leqslant n/2$ *и*

$$\frac{w_1\left[R^{2n-2r+1} + (w_1+1)^{2n-2r+1}\right] + R^{2n-2r}(w_1+1)^{2r-n-1}\left[R^2 - (w_1+1)^2\right]}{R + w_1 + 1},$$

если $r > n/2$.

Литература

[1] *Niederreiter H.* The linear complexity profile and the jump complexity of keystream sequences, I. Proceedings of Eurocrypt'90, Lecture notes in Comput. Sci., vol. 473, Springer, Berlin, 1991, p. 174–188.

[2] *Куракин В. Л.* Алгоритм Берлекемпа—Месси над конечными кольцами, модулями и бимодулями // Дискретная математика. 1998. Т. 10, № 4. С. 3–34.

Эквивалентные подпространства кода Рида—Маллера и множество открытых ключей криптосистемы Мак-Элиса—Сидельникова

И. В. Чижов

Кодовая криптосистема Мак-Элиса—одна из старейших криптосистем с открытым ключом. Она была предложена в 1978 году Р. Дж. Мак-Элисом [1]. В. М. Сидельников в работе [2] рассматривал криптосистему Мак-Элиса, построенную на основе двоичных кодов Рида—Маллера $\mathrm{RM}(r, m)$. В. М. Сидельников, проведя криптоанализ такого варианта криптосистемы Мак-Элиса, пришёл к выводу, что на сегодняшний день модификация криптосистемы Мак-Элиса, построенная на кодах Рида—Маллера, не обеспечивает необходимого уровня стойкости. В той же работе [2] автор предлагает усиленный вариант криптосистемы на основе кодов Рида—Маллера — криптосистему Мак-Элиса—Сидельникова.

Опишем устройство криптосистемы Мак-Элиса—Сидельникова. Пусть R — $(k \times n)$-порождающая матрица кода Рида—Маллера $\mathrm{RM}(r, m)$. Будем полагать, что матрица R состоит из векторов значений всех мономов от m переменных степени нелинейности, не выше r. К тому же, если перенумеровать сверху строки матрицы R, в строке с номером $2 \leqslant i \leqslant k$ стоит вектор значений монома $x_{i_1} x_{i_2} \ldots x_{i_t}$, где i_j $(1 \leqslant j \leqslant t)$ — номера позиций, в которых стоит единица в двоичном разложении числа $i - 1$, а в строке с номером 1 — вектор из всех единиц. Секретным ключом криптосистемы является кортеж

$$(H_1, H_2, \ldots, H_u, \Gamma).$$

Здесь $H_1, H_2, \ldots, H_u$ — невырожденные $(k \times k)$-матрицы над полем $F_2 = \{0, 1\}$, которые выбираются случайно и равновероятно из множества $\mathrm{GL}_k(F_2)$ всех двоичных невырожденных $(k \times k)$-матриц над полем F_2. Матрица Γ — перестановочная $(u \cdot n \times u \cdot n)$-матрица.

Открытым ключом криптосистемы Мак-Элиса—Сидельникова является матрица

$$G' = (H_1 R \parallel H_2 R \parallel \ldots \parallel H_u R) \cdot \Gamma,$$

где символом $\parallel$ обозначена конкатенация матриц по столбцам. Алгоритмы шифрования и расшифрования подробно описаны в [2].

Группу автоморфизмов кода Рида—Маллера $\mathrm{RM}(r, m)$ будем обозначать символом $\mathrm{Aut}(\mathrm{RM}(r, m))$. Рассмотрим некоторую перестановку P из группы автоморфизмов кода Рида—Маллера $\mathrm{RM}(r, m)$. Построим по ней i «длинных» перестановок $\mathcal{P}[i] \in S_{un}$ следующим образом: $\mathcal{P}[i](j) = j$ для любого $j \notin I = \{(i-1)\cdot n + 1, (i-1)\cdot n + 2, \ldots, i\cdot n\}$, а $\mathcal{P}[i](I) = P(I)$. Другими слова перестановка $\mathcal{P}[i]$ в i-том блоке совпадает с перестановкой P, а во всех остальных блоках — совпадает с единичной перестановкой. *Группой расширенных автоморфизмов* $\mathcal{A}_u(\mathrm{RM}(r, m))$ кода Рида—Маллера $\mathrm{RM}(r, m)$ назовём множество всех возможных произведений описанных перестановок $\mathcal{P}[i]$ для всех возможных $1 \leqslant i \leqslant u$ и всех возможных перестановок P из группы автоморфизмов кода Рида—Маллера.

Два секретных ключа $(H_1, H_2, \ldots, H_u, \Gamma)$ и $(H'_1, H'_2, \ldots, H'_u, \Gamma')$ назовём *эквивалентными*, если соответствующие им открытые ключи совпадают, то есть выполняется соотношение

$$(H_1R \parallel H_2R \parallel \ldots \parallel H_uR)\cdot\Gamma = (H'_1R \parallel H'_2R \parallel \ldots \parallel H'_uR)\cdot\Gamma'.$$

Всё множество секретных ключей разбивается на классы эквивалентности и число классов эквивалентности совпадает с числом открытых ключей. Класс эквивалентности с представителем $(H_1, \ldots, H_u, \Gamma)$ будем обозначать так: $[(H_1, \ldots, H_u, \Gamma)]$.

Рассмотрим множество $\mathcal{G}(H_1, H_2, \ldots, H_u)$, состоящее из перестановок $\Gamma \in S_{un}$, для которых существуют невырожденные двоичные матрицы H'_1, H'_2, ..., H'_u такие, что $(H_1R \parallel H_2R \parallel \ldots \parallel H_uR)\Gamma = (H'_1R \parallel H'_2R \parallel \ldots \parallel H'_uR)$.

Справедлива следующая теорема.

Теорема 1. *Класс эквивалентности* $[(H_1, H_2, \ldots, H_u, \Gamma)]$ *состоит из ключей вида*

$$(H'_1, H'_2, \ldots, H'_u, \Gamma_g\cdot\Gamma),$$

где Γ_g^{-1} *принадлежит множеству* $\mathcal{G}(H_1, H_2, \ldots, H_u)$ *и*

$$(H_1R \parallel H_2R \parallel \ldots \parallel H_uR) = (H'_1R \parallel H'_2R \parallel \ldots \parallel H'_uR)\Gamma_g.$$

Тем самым вопрос изучения эквивалентных секретных ключей сводится к описанию множеств $\mathcal{G}(H_1, \ldots, H_u)$.

Наиболее просто устроено множество $\mathcal{G}(E, \ldots, E)$. Следующее утверждение в немного другом виде было получено Г. А. Карпуниным [4].

Теорема 2. *Множество* $\mathcal{G}(E, \ldots, E)$ *состоит из перестановок вида* $\Gamma\cdot\mathcal{P}$, *где* Γ *такова, что*

$$(R \parallel \ldots \parallel R)\Gamma = (R \parallel \ldots \parallel R),$$

а $\mathcal{P}$ *— перестановка из группы* $\mathcal{A}_u(\mathrm{RM}(r, m))$.

Также в случае произвольного u справедлива следующая теорема.

Теорема 3. *Пусть для невырожденных матриц $D_1, D_2, \ldots, D_u$ существуют такие перестановки P_i $(1 \leqslant i \leqslant n)$ из S_n, что*

$$D_1 R = R \cdot P_1,\ D_2 R = R \cdot P_2,\ \ldots,\ D_u R = R \cdot P_u.$$

Обозначим через $\mathcal{P}_1[1], \mathcal{P}_2[2], \ldots, \mathcal{P}_u[u]$ перестановки из $\mathcal{A}_u(\mathrm{RM}(r, m))$ соответствующие перестановкам $P_1, P_2 \ldots, P_u$. И пусть H — любая невырожденная матрица. Тогда

$$\mathcal{G}(E, \ldots, E) = \mathcal{P}_1[1] \cdot \mathcal{P}_2[2] \ldots \mathcal{P}_u[u] \cdot \mathcal{G}(HD_1, \ldots, HD_u).$$

С учетом теоремы 1 теорема 3 дает описание нетривиального подмножества множества открытых ключей криптосистемы Мак-Элиса—Сидельникова в случае $u = 2$.

Обратимся к вопросу изучения множества $\mathcal{G}(H_1, H_2)$, то есть к изучению множества в случае двух блоков $(u = 2)$.

Для случая $u = 2$ задача поиска эквивалентных ключей криптосистемы Мак-Элиса—Сидельникова основывается на изучении множеств $\mathcal{G}(E, H)$. В случае когда матрица H задаёт автоморфизм кода $\mathrm{RM}(r, m)$ описание множества $\mathcal{G}(E, H)$ даёт теорема 3. Интересно получить описание этого множества в случае каких-то других матриц H.

Для некоторого вектора $\widetilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$, $\alpha_i = 1$, рассмотрим матрицу $T^i_{\widetilde{\alpha}}$ вида

$$T^i_{\widetilde{\alpha}} = \begin{array}{c} \\ \\ i \to \end{array} \begin{array}{c} \begin{array}{cccccc} & & & \overset{i}{\downarrow} & & \end{array} \\ \begin{pmatrix} 1 & 0 & \ldots & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots & \ldots & \vdots \\ \alpha_1 & \alpha_2 & \ldots & 1 & \ldots & \alpha_{k-1} \\ \vdots & \vdots & \ldots & \vdots & \ldots & \vdots \\ 0 & 0 & \ldots & 0 & \ldots & 1 \end{pmatrix} \end{array}$$

Первый случай — $i = 1$. Справедлива теорема.

Теорема 4. *Пусть $i = 1$. Тогда*

$$\mathcal{G}\left(E, T^1_{\widetilde{\alpha}}\right) = \left\{\Gamma \in \mathcal{G}(E, E) \;\middle|\; \left(0 \parallel (e^1 \oplus \widetilde{\alpha})R\right)\Gamma \in \mathrm{RM}(r, m) \times \mathrm{RM}(r, m)\right\}.$$

Здесь символом e^1 обозначен вектор, у которого на первом месте стоит 1, а на всех остальных местах — 0.

Перейдём теперь к изучению случая $i > 1$. Следует отметить, что в дальнейшем рассматриваются коды Рида—Маллера $\mathrm{RM}(r, m)$ с $r > 1$, так как

в случае $r = 1$ полное описание множества открытых ключей для двух блоков получено Г. А. Карпуниным [4]. При изучении эквивалентных ключей в случае $i > 1$ возникает необходимость в исследовании эквивалентности некоторых специальных $(k-1)$-мерных подпространств кода Рида—Маллера $\mathrm{RM}(r, m)$. Это объясняется тем, что справедливо следующее утверждение.

Утверждение 1. *Пусть R — порождающая матрица кода Рида—Маллера $\mathrm{RM}(r, m)$, $r > 1$. И пусть перестановка Γ с некоторыми невырожденными двоичными матрицами X, Y является решением уравнения*

$$(R \parallel T^i_{\widetilde{\alpha}} R)\Gamma = (XR \parallel YR),$$

то есть $\Gamma \in \mathcal{G}(E, T^i_{\widetilde{\alpha}})$. Тогда найдутся две перестановки $\Gamma' \in S_{un}$ и $\mathcal{P} = (\sigma_L \parallel \sigma_R) \in S_{un}$, σ_L, $\sigma_R \in S_n$ такие, что $\Gamma = \Gamma'\mathcal{P}$ и $(R' \parallel R')\Gamma' = (R' \parallel R')$. Здесь R' — матрица, получающаяся из матрицы R выкидыванием i-той строки.

Утверждение 1 сводит задачу изучения пространства эквивалентных ключей криптосистемы Мак-Элиса—Сидельникова, в случае $r > 1$, к изучении таких перестановок σ, что R'^{σ} — $(k-1)$-мерный подкод кода Рида—Маллера $\mathrm{RM}(r, m)$.

Для подкода R' введём ещё одно множество перестановок $I(R')$

$$I(R') = \{\delta \in S_n \mid \forall\, x \in R' \;\; x^{\delta} = x\}.$$

Иными словами множество $I(R')$ состоит из перестановок, которые не изменяют кодовые слова кода R'.

Теорема 5. *Пусть $2r \leqslant m$, $t > 0$. Тогда любая перестановка σ, для которой существует подпространство P кода Рида—Маллера такое, что*

$$R'^{\sigma} = P,$$

может быть представлена в виде произведения

$$\sigma = \delta \cdot \gamma,$$

где γ — аффинная перестановка, а δ принадлежит группе $I(R')$.

Используя теорему 5 можно получить описание множества $\mathcal{G}(E, T^i_{\widetilde{\alpha}})$ для $i > 1$.

Итак, справедлива следующая теорема 6.

Теорема 6. *Пусть $\mathrm{RM}(r, m)$ — код Рида—Маллера такой, что $2r \leqslant m$, i — натуральное число, большее единицы, H — любая невырожденная двоичная матрица, α — произвольный двоичный вектор,*

у которого в координате с номером i стоит единица. Тогда множество $\mathcal{G}\left(H,\, HT^i_{\widetilde{\alpha}}\right)$ *содержит те и только те перестановки* Γ, *которые могут быть представлены в виде*

$$\Gamma = \Gamma' \cdot (\sigma_L \parallel \sigma_R),$$

где σ_L, σ_R — *автоморфизмы кода Рида—Маллера* $\mathrm{RM}(r,\, m)$, *а для перестановки* Γ' *выполняются следующие два условия*:

1) *если* R' — $((k-1)\times n)$-*матрица, получающаяся выкидыванием строки с номером* i *из матрицы* R, *то*

$$(R' \parallel R')\Gamma' = (R' \parallel R');$$

2) *если* r^i — *строка матрицы* R *с номером* i, *то*

$$\left(r^i \parallel \widetilde{\alpha}R\right)\Gamma' \in \mathrm{RM}(r,\, m) \times \mathrm{RM}(r,\, m).$$

Литература

[1] *McEliece R. J.* A public-key cryptosystem based on algebraic coding theory // DSN Prog. Rep., Jet Prop. Lab., California Inst. Technol., Pasadena, CA, January 1978, p. 114–116.

[2] *Сидельников В. М.* Открытое шифрование на основе двоичных кодов Рида—Маллера // Дискретная математика. 1994. Т. 6, вып. 3. С. 3–20.

[3] *Мак-Вильямс Ф. Дж.*, *Слоэн Н. Дж. А.* Теория кодов, исправляющих ошибки. М.: Связь, 1979.

[4] *Карпунин Г. А.* О ключевом пространстве криптосистемы Мак-Элиса на основе двоичных кодов Рида—Маллера // Дискретная математика. 2004. Т. 16, вып. 2. С. 79–84.

Неравенства для ортогональных массивов большой силы

А. В. Халявин

1. Введение

Ортогональным массивом с N строками, n факторами, над алфавитом из s символов силы m называется таблица $N \times n$ с элементами из алфавита, в которой при выборе любых m столбцов, любая из s^m комбинаций символов в этих столбцах встречается среди строк одинаковое число раз. Для ортогональных массивов принято краткое обозначение OA(N, n, s, m). При этом пустые массивы не рассматриваются. Если все строки массива различны, то он называется *простым*. Ортогональным массивам целиком посвящена монография [4].

Ортогональные массивы тесно связаны с корреляционно-иммунными функциями. Под *корреляционно-иммунной* булевой функцией порядка m понимается булева функция, у которой доля единичных значений не меняется при подстановке констант вместо любых m переменных. Набор аргументов, на которых корреляционно-иммунная функция порядка m принимает единичные значения, образует ортогональный массив силы m. Аналогично можно наоборот сопоставить корреляционно-иммунную функцию каждому простому ортогональному массиву. Однако, в теории булевых функций, в отличие от теории ортогональных массивов, имеет большую важность свойство уравновешенности. Булева функция называется *уравновешенной*, если она принимает единицу ровно на половине наборов. Уравновешенные корреляционно-иммунные функции порядка m называются *m-устойчивыми*. Подробнее о корреляционно-иммунных функциях можно прочитать в [2].

Представляет интерес вопрос, при каких соотношениях между n и m существуют неуравновешенные неконстантные корреляционно-иммунные функции порядка m от n переменных. Примеры таких функций при $m =$

Работа выполнена при поддержке гранта Президента РФ (проект НШ 4470.2008.1) и программы фундаментальных исследований ОМН РАН «Алгебраические и комбинаторные методы математической кибернетики» (проект «Синтез и сложность управляющих систем»).

$= 2n/3 - 1$ хорошо известны. В работе [3] Д. Г. Фон дер Флаасс доказал, что при $m \geqslant (2n-2)/3$ любая неконстантная корреляционно-иммунная функция порядка m от n переменных является уравновешенной. Данная статья обобщает этот факт на случай не обязательно простых ортогональных массивов.

2. Основной результат

Теорема. *Если* $m \geqslant (2n-2)/3$, *то для* $\mathrm{OA}(N, n, 2, m)$ *выполнено* $N \geqslant 2^{n-1}$. *А если при этом* $N = 2^{n-1}$, *то ортогональный массив является простым.*

Доказательство. Для $\alpha \in F_2^n$ обозначим $x_\alpha = 2n_\alpha - 1$, где n_α — число строк ортогонального массива, совпадающих с α. Предположим, что $0 < N \leqslant 2^{n-1}$. Тогда

$$-2^n < \sum_\alpha x_\alpha = 2N - 2^n \leqslant 0. \tag{1}$$

Применим к набору чисел $\{x_\alpha\}$ преобразование Уолша:

$$W_\beta = \sum_\alpha x_\alpha (-1)^{(\alpha,\beta)},$$

для которого справедлива формула обращения [1]:

$$x_\alpha = \frac{1}{2^n} \sum_\beta W_\beta (-1)^{(\alpha,\beta)}. \tag{2}$$

Возведем теперь выражение x_α через W_α в квадрат. Получим

$$x_\alpha^2 = \frac{1}{2^{2n}} \sum_{\beta,\gamma} W_\beta W_\gamma (-1)^{(\alpha,\beta)+(\alpha,\gamma)} = \frac{1}{2^n} \sum_\beta \Phi_\beta (-1)^{(\alpha,\beta)}, \tag{3}$$

где

$$\Phi_\alpha = \frac{1}{2^n} \sum_{\beta,\gamma,\beta+\gamma=\alpha} W_\beta W_\gamma. \tag{4}$$

Поскольку ортогональный массив имеет силу m, то $W_\alpha = 0$ при $0 < |\alpha| \leqslant m$. Если $|\beta| \geqslant m+1$ и $|\gamma| \geqslant m+1$, то $|\beta+\gamma| \leqslant 2(n-m-1) \leqslant m$. Поэтому при $|\alpha| > m$ мы получаем

$$\Phi_\alpha = \frac{W_0}{2^{n-1}} W_\alpha. \tag{5}$$

Кроме того (используем формулу обращения для (3) и определение x_α),

$$\Phi_0 = \sum_\alpha x_\alpha^2 \geqslant \sum_\alpha 1 = 2^n. \tag{6}$$

Рассмотрим теперь величину $\sum\limits_{\alpha} x_\alpha^3$. Это нулевой коэффициент Уолша у произведения векторов x_α и x_α^2, поэтому аналогично выражению (4), перемножая формулы (2) и (3), получим равенство

$$\sum_{\alpha} x_\alpha^3 = \frac{1}{2^n}\sum_{\alpha}\Phi_\alpha W_\alpha = \frac{1}{2^n}\left(\Phi_0 W_0 + \sum_{\alpha,|\alpha|\leqslant m}\Phi_\alpha W_\alpha + \sum_{\alpha,|\alpha|>m}\Phi_\alpha W_\alpha\right). \qquad (7)$$

Второе слагаемое равно 0, поскольку $W_\alpha = 0$, а в третьем слагаемом можно подставить выражение (5) для Φ_α. Если, кроме того, вычесть $\sum\limits_{\alpha} x_\alpha$, то (7) преобразуется к виду

$$\begin{aligned}\sum_{\alpha}(x_\alpha^3 - x_\alpha) &= \frac{1}{2^n}\left(\Phi_0 W_0 + \frac{2}{2^n}\sum_{\alpha,|\alpha|>m} W_0 W_\alpha^2\right) - W_0 = \\ &= \frac{W_0}{2^n}\left(\Phi_0 - \frac{2}{2^n}W_0^2 + \frac{2}{2^n}\sum_{\alpha} W_\alpha^2 - 2^n\right) = \\ &= \frac{W_0}{2^n}\left(3\Phi_0 - \frac{2}{2^n}W_0^2 - 2^n\right).\end{aligned}$$

Оценим сомножители. Поскольку $W_0 \leqslant 0$, то и $W_0/2^n \leqslant 0$. Из (6) и (1) получаем $\Phi_0 \geqslant 2^n \geqslant W_0^2/2^n$, откуда $3\Phi_0 - (2/2^n)W_0^2 - 2^n \geqslant 0$. Таким образом, $\sum\limits_{\alpha}(x_\alpha^3 - x_\alpha) \leqslant 0$. С другой стороны, $x_\alpha^3 - x_\alpha \geqslant 0$, поскольку x_α может принимать лишь значения -1, 1, 3, ... Отсюда получаем, что $x_\alpha^3 - x_\alpha = 0$ для всех α, то есть $x_\alpha = \pm 1$ и, как следствие, массив является простым. Кроме того, получаем, что либо $W_0 = 0$, либо $3\Phi_0 - (2/2^n)W_0^2 - 2^n = 0$. В первом случае ортогональный массив содержит $\sum\limits_{\alpha} n_\alpha = \sum\limits_{\alpha}(x_\alpha + 1)/2 = W_0/2 + 2^{n-1} = 2^{n-1}$ строк, что удовлетворяет заключению теоремы. Во втором случае $\Phi_0 = 2^n$ и $|W_0| = 2^n$, что противоречит неравенствам (1). □

Литература

[1] *Логачёв О. А.*, *Сальников А. А.*, *Ященко В. В.* Булевы функции в теории кодирования и криптографии. М.: МЦНМО, 2004.

[2] *Тарannиков Ю. В.* О корреляционно-иммунных и устойчивых булевых функциях // Математические вопросы кибернетики. Вып. 11. М.: Физматлит, 2002. С. 91–148.

[3] *Fon-Der-Flaass D. G.* A bound on correlation immunity // Siberian Electronic Mathematical Reports (`http://semr.math.nsc.ru/`). 2007. V. 4. P. 133–135.

[4] *Hedayat A. S.*, *Sloane N. J. A.*, *Stufken J.* Orthogonal arrays: theory and applications. New York: Springer-Verlag, 1999.

Теоретико-сложностной подход к оценке сложности решения систем булевых уравнений

С. П. Горшков, А. В. Тарасов

Необходимость решения систем булевых уравнений возникает в криптографии, теории кодирования, теории конечных автоматов. Основная задача, возникающая при решении систем булевых уравнений, состоит в построении методов решения, имеющих по возможности наименьшую временную сложность.

Продолжительный анализ некоторых классов систем уравнений не привел к построению для этих классов полиномиальных алгоритмов решения. По всей видимости, для этих классов систем вообще не существует полиномиальных алгоритмов решения, но современное состояние математики позволяет доказывать такого рода результаты только при условии справедливости гипотезы $\mathrm{P} \neq \mathrm{NP}$. При этом используется теория NP-полных задач [4].

В работе рассматриваются классы систем булевых уравнений с ограничениями на выбор функций и без ограничений на выбор неизвестных [7] (классы систем вида $[F]_{\mathrm{NC}}$). В зарубежной литературе изложение ведется в терминах булевых формул, при этом задача распознавания совместности указанных систем уравнений называется «Constraint satisfaction problems» [18]. Если F — набор булевых функций, то эта задача обозначается $\mathrm{SAT}(F)$ (задача $\mathrm{SAT}(F)$ эквивалентна рассматриваемой далее задаче $\mathrm{SAT}([F]_{\mathrm{NC}})$). В ряде работ слабо отрицательные булевы функции называются хорновскими функциями, а слабо положительные булевы функции — антихорновскими функциями.

Среди зарубежных работ по рассматриваемой тематике следует выделить обзор [18] и монографию [21], поскольку в них приведены все основные результаты, полученные в англоязычной литературе до 2001 года. Новые продвижения даны в обзоре [22]. Среди отечественных работ по данной тематике основными являются, по-видимому, следующие статьи: [7, 6, 9]. Часть исследований в рассматриваемой области осуществлена в рамках НИР, проводимых Академией криптографии Российской Федерации.

Введем некоторые обозначения и определения.

- N — множество натуральных чисел, $N_0 = N \cup \{0\}$;
- B_n — n-мерное пространство булевых векторов, $n \in N_0$.

Определение 1 ([34]). Булева функция $f(x_1, \ldots, x_k)$ называется

1) 0-*выполнимой*, если $f(0, \ldots, 0) = 1$;
2) 1-*выполнимой*, если $f(1, \ldots, 1) = 1$;
3) *мультиаффинной*, если существует представление функции f в виде конъюнкции аффинных функций:

$$f(x_1, \ldots, x_k) = \bigwedge_{i=1}^{t} (\alpha_{i1} \cdot x_1 \oplus \ldots \oplus \alpha_{ik} \cdot x_k \oplus \alpha_{i_0}); \qquad (1)$$

4) *биюнктивной*, если существует представление f в виде 2-КНФ:

$$f(x_1, \ldots, x_k) = \bigwedge_{i=1}^{t} (x_{s_{i1}}^{a_{i1}} \vee x_{s_{i2}}^{a_{i2}}); \qquad (2)$$

5) *слабо положительной*, если существует представление f в виде следующей КНФ:

$$f(x_1, \ldots, x_k) = \bigwedge_{i=1}^{t} (x_{s_{i1}}^{a_i} \vee x_{s_{i2}} \vee \ldots \vee x_{s_{ir_i}}); \qquad (3)$$

6) *слабо отрицательной*, если существует представление f в виде следующей КНФ:

$$f(x_1, \ldots, x_k) = \bigwedge_{i=1}^{t} (x_{s_{i1}}^{a_i} \vee \overline{x}_{s_{i2}} \vee \ldots \vee \overline{x}_{s_{ir_i}}); \qquad (4)$$

7) 2-*монотонной*, если f представима в виде ДНФ одного из трех видов: $x_{i_1} \ldots x_{i_p}$, $\overline{x}_{j_1} \ldots \overline{x}_{j_q}$, $x_{i_1} \ldots x_{i_p} \vee \overline{x}_{j_1} \ldots \overline{x}_{j_q}$, $\{i_1, \ldots, i_p\} \cap \cap \{j_1, \ldots, j_q\} = \varnothing$.

Множество всех функций соответственно классов 1–7 обозначим 0-S, 1-S, A, Bi, WP, WN, 2-M. Формулы (1), (2), (3), (4), соответственно, для функций классов A, Bi, WP, WN будем называть *приведенными представлениями*. Множество A $\cap$ Bi, представляющее собой множество мультаффинных функций, являющихся произведениями аффинных функций от не более чем 2-х переменных, обозначим через 2-A.

Определение 2 ([7]). Пусть $F = \{f_j(x_1, \ldots, x_{k_j}) \mid j \in J\}$ — некоторый набор булевых функций, $X = \{x_i \mid i \in N\}$. Символом $[F]_{\mathrm{NC}}$ обозначим класс всевозможных систем уравнений, каждое уравнение которых имеет следующую структуру: правая часть равна 1, левая часть есть функция из набора F, переменные выбираются из множества X; другие ограничения на системы класса $[F]_{\mathrm{NC}}$ не накладываются. Так, если $F = \{f_j(x_1, \ldots, x_k) \mid j = 1, \ldots, d\}$ — конечный набор булевых функций, то системы класса $[F]_{\mathrm{NC}}$ имеют вид:

$$f_{s_j}(x_{s_{i1}}, \ldots, x_{s_{ik}}) = 1, \quad i = 1, \ldots, m,$$

где $m \in N$, $f_{s_i} \in F$, $x_{s_{ij}} \in X$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, k$.

Классы систем уравнений вида $[F]_{\mathrm{NC}}$ будем называть *классами систем булевых уравнений с ограничениями на выбор функций и без ограничений на выбор неизвестных*. При этом набор функций F будем называть *порождающим набором функций* для класса систем $[F]_{\mathrm{NC}}$.

Пусть R — некоторый класс систем булевых уравнений, S — система уравнений из класса R, $\mathrm{sol}(S)$ — множество решений системы S, $\mathrm{len}(S)$ — длина записи системы S. Определим две задачи, связанные с решением систем уравнений.

Определение 3. 1. Задача $\mathrm{SAT}(R)$ распознавания совместности систем класса R.

Условие. Задана система $S \in R$.

Вопрос. Верно ли, что $|\mathrm{sol}(S)| > 0$?

2. Задача $\mathrm{ENU}(R)$ определения числа решений систем класса R.

Условие. Задана система $S \in R$.

Вопрос. Какова мощность множества $\mathrm{sol}(S)$?

Определение 4 ([7]). Класс систем уравнений R назовем *полиномиально решаемым*, если

1) задача $\mathrm{SAT}(R)$ полиномиальна;
2) существуют алгоритм A и полином $p(n)$ такие, что для всякой совместной системы $S \in R$ алгоритм A последовательно, без повторений находит все решения системы S, причем сложность получения первого и каждого очередного t-ого решения, после получения $(t-1)$-го решения, не более $p(\mathrm{len}(S))$;
3) в ходе работы алгоритму A требуется память не более $p(\mathrm{len}(S))$.

Заметим, что во всех зарубежных работах порождающий набор F считается конечным. При этом не требуется накладывать ограничения на вид записи функций набора F. В работах [7, 8] рассматривается, в том числе, случай бесконечного порождающего набора F.

Если набор функций F бесконечен, то считается, что функции набора F записаны или соответствующими приведенными представлениями, или многочленами Жегалкина, или ДНФ

Объединяя результаты теоремы разделимости для задач $\mathrm{SAT}([F]_{\mathrm{NC}})$ [34], теоремы разделимости для задач $\mathrm{SAT}([F]_{\mathrm{NC}})$ при исключении тривиальных решений, все координаты которых равны между собой [7] (в работе [18] приведен аналог этой теоремы для случая конечного порождающего набора F), теоремы разделимости для задач $\mathrm{ENU}([F]_{\mathrm{NC}})$ [5, 7] (в работе [19] доказан аналог этой теоремы для случая конечного порождающего набора F), и результаты о полиномиально решаемых классах систем булевых уравнений [7], получаем следующую общую картину, раскрывающую сложность решения систем уравнений из классов вида $[F]_{\mathrm{NC}}$.

Если все функции порождающего набора F являются мультиаффинными: $F \subset \mathrm{A}$ (системы класса $[F]_{\mathrm{NC}}$ равносильны системам линейных уравнений), то задачи $\mathrm{SAT}([F]_{\mathrm{NC}})$ и $\mathrm{ENU}([F]_{\mathrm{NC}})$ полиномиальны, класс систем уравнений $[F]_{\mathrm{NC}}$ является полиномиально решаемым.

В случае, когда $F \not\subset \mathrm{A}$, но выполняется хотя бы одно из включений $F \subset \mathrm{Bi}$, $F \subset \mathrm{WP}$, $F \subset \mathrm{WN}$, возникают следующие интересные соотношения: задача $\mathrm{SAT}([F]_{\mathrm{NC}})$ полиномиальна, класс систем уравнений $[F]_{\mathrm{NC}}$ является полиномиально решаемым, а задача $\mathrm{ENU}([F]_{\mathrm{NC}})$ является #P-полной (труднорешаемой).

Если же $F \not\subset \mathrm{A}$, $F \not\subset \mathrm{Bi}$, $F \not\subset \mathrm{WP}$, $F \not\subset \mathrm{WN}$, то задача выяснения существования нетривиальных решений систем класса $[F]_{\mathrm{NC}}$ является NP-полной, задача определения числа решений — #P-полная, класс систем уравнений $[F]_{\mathrm{NC}}$ не является полиномиально решаемым (в предположении $\mathrm{P} \neq \mathrm{NP}$).

Из приведенных результатов следует также следующий факт: в предположении $\mathrm{P} \neq \mathrm{NP}$ множество полиномиально решаемых классов систем булевых уравнений вида $[F]_{\mathrm{NC}}$ исчерпывается случаем выполнения хотя бы одного из включений

$$F \subset \mathrm{A}, \quad F \subset \mathrm{Bi}, \quad F \subset \mathrm{WP}, \quad F \subset \mathrm{WN}. \tag{5}$$

В статье [29] изучается так называемая обратная задача к задаче распознавания совместности систем вида $[F]_{\mathrm{NC}}$. На языке классов систем булевых уравнений с ограничениями на выбор функций и без ограничений на выбор неизвестных результаты работы выглядят следующим образом. Определим обратную задачу к задаче выполнимость (эту задачу обозначим $\mathrm{INVSAT}([F]_{\mathrm{NC}})$).

Условие. Дано натуральное число n и множество $M \subset B_n$.

Вопрос. Найдется ли система уравнений из класса $[F]_{\mathrm{NC}}$ такая, что множество ее решений равно M?

Теорема 1 (Теорема разделимости для задачи INVSAT($[F]_{\mathrm{NC}}$), [29]). *Если для набора функций F выполняется хотя бы одно из включений* (5), *то задача INVSAT($[F]_{\mathrm{NC}}$) имеет полиномиальную сложность, в противном случае эта задача является coNP-полной.*

Рассмотрим задачу «Выполнимость с кванторами» [23]. Пусть имеется класс систем $[F]_{\mathrm{NC}}$, S — некоторая система из этого класса, а формула $f(x_1, \ldots, x_k)$ есть произведение формул левой части системы S. Рассмотрим выражение

$$Q_1 x_1 \ldots Q_k x_k \; f(x_1, \ldots, x_k) \tag{6}$$

где Q_i — кванторы, $Q_i \in \{\forall, \exists\}$, $i = 1, \ldots, k$. В формуле (6) все переменные являются связанными и ставится вопрос: формула (6) является истиной или ложью? Эту задачу обозначим QSAT($[F]_{\mathrm{NC}}$).

Теорема 2 (Теорема разделимости для задачи QSAT($[F]_{\mathrm{NC}}$), [23]). *Если для конечного набора функций F выполняется хотя бы одно из включений* (5), *тогда задача QSAT($[F]_{\mathrm{NC}}$) является полиномиальной. В случае, когда не выполняется ни одно из включений* (5), *задача QSAT($[F]_{\mathrm{NC}}$) является PSPACE-полной.*

Рассмотрим теперь задачу выполнимости в оптимизационной постановке. В работах [21, 30, 28, 22] рассмотрено понятие задачи оптимизации и класса NP проблем оптимизации (NPO). Класс полиномиально решаемых задач из NPO обозначается через PO. К классу проблем оптимизации относятся задачи:

- MAXSAT($[F]_{\mathrm{NC}}$) — задача нахождения вектора, выполняющего максимальное количество уравнений системы;
- MINSAT($[F]_{\mathrm{NC}}$) — задача нахождения вектора, минимизирующего количество не выполненных уравнений системы (данная задача эквивалентна задаче MAXSAT($[F]_{\mathrm{NC}}$) в случае, когда обе эти задачи полиномиальны, но отличается от нее при приближенном решении);
- MAXONES($[F]_{\mathrm{NC}}$) — задача нахождения решения системы максимального веса;
- MINONES($[F]_{\mathrm{NC}}$) — задача нахождения решения системы минимального веса.

Данные задачи могут быть сформулированы и во «взвешенном» варианте. В этом случае каждому уравнению для задач MAXSAT и MINSAT, либо каждой координате решения для задач MAXONES и MINONES приписывается неотрицательный «вес» и задача решается относительно суммарного «веса». В этом случае задачи обозначаются Weighted MAXSAT($[F]_{\mathrm{NC}}$),

Weighted MINSAT($[F]_{NC}$), Weighted MAXONES($[F]_{NC}$) и Weighted MINONES($[F]_{NC}$). В случае, когда какой-либо результат верен как для не взвешенного, так и для взвешенного варианта задачи, слово Weighted будем писать в скобках, например (Weighted) MAXSAT($[F]_{NC}$), следуя вышеназванным работам.

Замечание. В работах [21, 28, 30], а также ряде других работ, оптимизационные задачи обозначаются как (Weighted) MAXSAT(F), (Weighted) MINSAT(F), (Weighted) MAXONES(F) и (Weighted) MINONES(F), что имеет тот же смысл, что и во введенных нами обозначениях.

Для задач оптимизации рассматриваются возможности приближенного решения задачи, то есть нахождения решения, в некотором смысле близкого к оптимальному. Введем понятие приближения для алгоритма решения задачи выполнимости в оптимизационной постановке (одной из четырех, указанных выше), которое согласуется с определением из работы [21]. Пусть $S \in [F]_{NC}$ — система уравнений от k неизвестных, $m(S, x)$ — значение целевой функции (то есть соответствующей суммы весов уравнений либо неизвестных) на наборе значений $x \in B_k$, $\mathrm{OPT}(S)$ — оптимальное значение целевой функции $m(S, x)$.

Определение 5 ([21]). Приближенный алгоритм A для решения NPO-задачи $\Pi \in \{$(Weighted) MAXSAT($[F]_{NC}$), (Weighted) MINSAT($[F]_{NC}$), (Weighted) MAXONES($[F]_{NC}$), (Weighted) MINONES($[F]_{NC}$)$\}$ имеет аппроксимацию $R(n)$, если для каждой системы S от k неизвестных, длина записи которой равна n, он находит вектор $x \in B_k$, удовлетворяющее соотношению:

$$\max\left\{\frac{m(S, x)}{\mathrm{OPT}(S)}, \frac{\mathrm{OPT}(S)}{m(S, x)}\right\} \leqslant R(n). \qquad (7)$$

Если полиномиальный алгоритм A имеет аппроксимацию 1, то он находит оптимальное решение. В этом случае он решает NPO-задачу с полиномиальной сложностью и задача оказывается в классе PO. В предположении P$\neq$NP для NP-полных задач таких алгоритмов не существует, однако для многих из них есть алгоритмы с приближением $1 + \varepsilon$ для любого $\varepsilon > 0$. В зависимости от того, какого вида функцию $R(n)$ в (7) можно выбрать, выделяется ряд классов задач оптимизации. Если существует такое $C > 0$, что $R(n) = C$, то задача лежит в классе APX. Если $R(n)$, то задача лежит в классе log-APX, а если $R(n)$ ограничено полиномом от n то задача лежит в классе poly-APX.

Для задач оптимизации определяются два основных типа сводимости: A-сводимость и AP-сводимость [21], в соответствии с которыми определяются понятия полноты. Данные сводимости определяют соответствия между системами из разных классов и устанавливают связь между значениями

целевых функций. В соответствии с введенными понятиями в [21, 22] определяются понятия APX-полноты, log-APX-полноты и poly-APX-полноты. В указанных работах сформулирована теорема разделимости для задачи (Weighted) MAXSAT($[F]_{\mathrm{NC}}$), в соответствии с которой эта задача либо полиномиальна, либо APX-полна, причем полиномиальность этой задачи исчерпывается случаем, когда выполняется одно из включений: $F \subset$ 0-S, $F \subset$ 1-S, $F \subset$ 2-M.

Классификация сложности для задачи (Weighted) MAXONES($[F]_{\mathrm{NC}}$) имеет более сложную структуру: задача (Weighted) MAXONES($[F]_{\mathrm{NC}}$) может быть полиномиальной, APX-полной, poly-APX-полной, разрешимой, но не аппроксимируемой, либо неразрешимой. В частности, если выполняется одно из включений $F \subset$ 1-S, $F \subset$ WP, $F \subset$ 2-A, то задача (Weighted) MAXONES($[F]_{\mathrm{NC}}$) — полиномиальна. Эта же задача APX-полна для $F \subset$ A, poly-APX-полна для случая $F \subset$ WN или $F \subset$ Bi. Если же $F \subset$ 0-S, то задача поиска выполняющего вектора полиномиальна, но задача поиска решения положительного веса уже NP-трудна. Наконец, во всех остальных случаях задача поиска любого выполняющего решения для (Weighted) MAXONES($[F]_{\mathrm{NC}}$) NP-трудна. В тех же работах [21, 22] получена полная классификация сложности задач минимизации (Weighted) MINSAT($[F]_{\mathrm{NC}}$) и (Weighted) MINONES($[F]_{\mathrm{NC}}$).

Одной из важных подзадач проблемы Weighted MAXSAT($[F]_{\mathrm{NC}}$), является задача решения систем булевых уравнений с искаженной правой частью, в которой фактически необходимо найти вектор с максимальным суммарным весом выполняемых уравнений. Понятие систем уравнений с искаженной правой частью и некоторые методы их решения рассмотрены в работах [2, 3].

Ряд работ посвящен исследованию свойств булевых функций, порождающих полиномиально решаемые классы систем уравнений $[F]_{\mathrm{NC}}$ — изучению мультиаффинных, биюнктивных, слабо положительных и слабо отрицательных булевых функций. В работах [34, 6] получены критерии мультиаффинности, биюнктивности, слабой положительности и слабой отрицательности булевых функций.

В [9] оценивается сложность задач распознавания мультиаффинности, биюнктивности, слабой положительности и слабой отрицательности при различных заданиях булевых функций, эти результаты важны при использовании отмеченных выше теорем разделимости (некоторые подобные результаты содержатся в монографии [21]). В работе [7] получены оценки числа мультиаффинных и биюнктивных функций, а в [1] получена оценка числа слабо отрицательных функций.

Свойства биюнктивных функций изучены также в работах [11, 13, 14]. В работе [13] показано, что задача минимизации 2-КНФ, представля-

ющей биюнктивную функцию является полиномиальной, в то время как в общем случае эта задача является труднорешаемой. Построен полиномиальный алгоритм минимизации 2-КНФ, представляющей биюнктивную функцию, описаны все минимальные 2-КНФ. В работе [14] исследована сложность распознавания эквивалентности биюнктивных функций, заданных своими 2-КНФ, относительно некоторых групп. Показано, что данная задача полиномиально эквивалентна задаче распознавания изоморфизма графов в случае симметрической группы и группы Джевонса и полиномиальна в случае группы сдвигов, рассмотрен ряд свойств групп инерции биюнктивных функций в данных группах. В статье [11] рассмотрена задача описания биюнктивных функций, инвариантных относительно заданной перестановки переменных. Задача полностью решена для полноцикловой подстановки.

Активно исследуются свойства слабо положительных и слабо отрицательных функций [15, 16, 24, 26, 31, 35]).

В работе [10] изучаются булевы функции, которые одновременно принадлежат двум или более классам из A, Bi, WP, WN. Асимптотическая формула для числа функций одного из возникающих подклассов булевых функций получена в работе [12].

В работах [17, 33] рассмотрена, в частности, следующая задача: имеются два набора функций F_1 и F_2, спрашивается — порождают они одинаковые классы систем уравнений $[F_1]_{\mathrm{NC}}$, $[F_2]_{\mathrm{NC}}$ или разные?

В работах [22, 32] сформулированы открытые проблемы в рассматриваемой области. В частности, является открытым вопрос о NP-трудности APX-полных задач MAXSAT.

Литература

[1] *Алексеев В. Б.* О числе семейств подмножеств, замкнутых относительно пересечения // Дискретная математика, 1989, т. 1, вып. 2, с. 129–136.

[2] *Балакин Г. В.* Введение в теорию случайных систем уравнений // Труды по дискретной математике, 1997, т. 1, с. 1–18.

[3] *Балакин Г. В.* Алгоритм нахождения множества наименьшей мощности, содержащего истинное решение с заданной вероятностью // Труды по дискретной математике, 2003, т. 7, с. 7–21.

[4] *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. Пер. с англ. // М.: Мир, 1982, 416 с.

[5] *Горшков С. П.* О сложности нахождения числа выполняющих наборов значений переменных в задаче «Обобщенная выполнимость» // Материалы девятой Всесоюзной конференции по математической логике, 1988, с. 38.

[6] *Гизунов С. А.*, *Носов В. А.* О классификации всех булевых функций четырех переменных по классам Шефера // Обозрение прикладной и промышленной математики. Серия «Дискретная математика», 1995, т. 2, вып. 3, с. 440–467.

[7] *Горшков С. П.* Применение теории NP-полных задач для оценки сложности решения систем булевых уравнений // Обозрение прикладной и промышленной математики. Серия «Дискретная математика», 1995, т. 2, вып. 3, с. 325–398.

[8] *Горшков С. П.* О сложности задачи нахождения числа решений систем булевых уравнений // Дискретная математика, 1996, т. 8, вып. 1, с. 72–85.

[9] *Горшков С. П.* О сложности распознавания мультиаффинности, биюнктивности, слабой положительности и слабой отрицательности булевых функций // Обозрение прикладной и промышленной математики. Серия «Дискретная математика», 1997, т. 4, вып. 2, с. 216–237.

[10] *Горшков С. П.* О пересечении классов мультиаффинных, биюнктивных, слабо положительных и слабо отрицательных булевых функций // Обозрение прикладной и промышленной математики. Серия «Дискретная математика», 1997, т. 4, вып. 2, с. 238–259.

[11] *Ролдугин П. В.*, *Тарасов А. В.* О числе биюнктивных функций, инвариантных относительно данной подстановки // Дискретная математика, 2002, т. 14, вып. 3, с. 23–41.

[12] *Сачков В. Н.* Разбиения с поглощениями и противоречивые разбиения множеств // Труды по дискретной математике, 2001, т. 4, с. 201–222.

[13] *Тарасов А. В.* О свойствах функций, представимых в виде 2-КНФ // Дискретная математика, 2001, т. 13, вып. 4, с. 99–115.

[14] *Тарасов А. В.* Некоторые свойства групп инерции булевых биюнктивных функций и индуктивный метод генерации таких функций // Дискретная математика, 2002, т. 14, вып. 2, с. 34–47.

[15] *Arvind V.*, *Biswas S.* An $O(n^2)$ algorithm for the satisfiability problem of a subset of propositional sentences in CNF that includes all Horn sentences // Information Processing Letters, 1987, v. 24, no. 1, p. 67–69.

[16] *Angluin D.*, *Frazier M.*, *Pitt L.* Learning conjunctions of Horn clauses // Proceedings of the 31st Annual Symposium on Foundations of Computer Science, 1990, p. 186–192.

[17] *Bohler E.*, *Hemaspaandra E.*, *Reith S.*, *Vollmer H.* Equivalence problems for Boolean constraint satisfaction // Preprint, Reihe Institut fur Informatik Universitat Wurzburg, 2001, № 282, 16 p.

[18] *Creignou N.*, *Hermann M.* Complexity of constraint satisfaction problems // Survey Document for CP 2001 Tutorial, 2001, 33 p.

[19] *Creignou N.*, *Hermann J.* Complexity of generalized satisfiability counting problems // Information and Computation, 1996, v. 125, no. 1, p. 1–12.

[20] *Creignou N.*, *Hebrard J.* On generating all satisfying truth assignments of a generalized CNF-formula // Theoretical Informatics and Applications, 1997, v. 31, no. 6, p. 499–511.

[21] *Creignou N.*, *Khanna S.*, *Sudan M.* Complexity classifications of Boolean constraint satisfaction problems // SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, 2001, 106 p.

[22] *Creignou N.* Boolean CSP // Universite de la Mediterranee, 2006, 82 p.

[23] *Dalmau V.* Some dichotomy theorems on quantified Boolean formulas // Technical Report LSI-97-43-R, Departament LSI, Universitat Politecnica de Catalunya, 1997, 23 p.

[24] *Dowling W. F.*, *Gallier J. H.* Linear-time algorithms for testing the satisfiability of propositional Horn formulae // J. Logic Programming, 1984, no. 3, p. 267–284.

[25] *Furer M.*, *Kasiviswanathan S. P.* Algorithms for counting 2-SAT solutions and colouring with applications // Electronic colloquium on computational complexity, 2007.

[26] *Hammer P.*, *Kogan A.* Horn functions and their DNFs // Information Processing Letters, 1992, v. 44, no. 1, p. 23–29.

[27] *Jonnson D.*, *Yannakakis M.*, *Papadimitriou C.* On generating all maximal independent sets // Information Processing Letters, 1988, v. 27, no. 3, p. 119–123.

[28] *Kolaitis G.* Constraint satisfaction, databases and logic // 2004, p. 1587–1595.

[29] *Kavvadias D.*, *Sideri M.* The inverse satisfiability problem // SIAM J. on Computing, 1998, v. 28, no. 3, p. 152–163.

[30] *Khanna S.*, *Sudan M.*, *Trevisan L.*, *Williamson D.* The approximability of constraint satisfaction problems // SIAM J. on Computing, 2001, v. 30, no. 6, p. 1863–1920.

[31] *Minoux M.* LTUR: a simplified linear-time unit resolution algorithm for Horn formulae and computer implementation // Information Processing Letters, 1988, v. 29, no. 1, p. 1–12.

[32] Open Problems List. Arising from MathsCSP, Workshop, Oxford, March 2006, Version 0.3, April 2006.

[33] *Reith S.* Generalized satisfiability problems // Dissertation zur Erlangung des naturwissenschaftlichen Doktorgrades der Bayerischen Julius-Maximilians-Universitat Wurzburg, 2001, 103 p.

[34] *Schaefer T.* Complexity of satisfiability problems // Proceedings of the 10th Annual ACM Symposium on theory of computing machinery, 1978, p. 216–226.

[35] *Yamasaki S.*, *Doshita S.* The satisfiability problem for a class consisting of Horn sentences and non-Horn sentences in proportional logic // Information and Control, 1983, v. 59, no. 1–3, p. 1–12.

Линейный алгоритм, определяющий по вектору значений булевой функции, задается ли она полиномом фиксированной степени

С. Н. Селезнёва

Каждая булева функция однозначно задается полиномом по mod 2. Степенью булевой функции называется степень задающего ее полинома. В криптографии важную роль играют булевы функции фиксированных степеней, например, степени 1 или 2.

Пусть булева функция записана вектором своих значений с $N = 2^n$ координатами, где n — число переменных функции. Известно, что по вектору значений булевой функции ее полином можно найти со сложностью $O(N \log N)$ [1]. Поэтому при отыскании алгоритмов, распознающих свойства полиномов булевых функций по их векторам значений, имеет смысл рассматривать только алгоритмы, имеющие меньшую по порядку сложность.

В настоящей заметке предлагается линейный по сложности алгоритм, который определяет по вектору значений булевой функции, задается ли она полиномом фиксированной степени, и в случае положительного ответа строит этот полином.

Пусть $B = \{0, 1\}$, V^n — множество векторов длины n с координатами из множества B. *Булевой функцией* назовем отображение

$$f^n : V^n \to B, \quad n = 0, 1, \ldots.$$

Множество всех булевых функций, зависящих от n переменных, обозначим как $\mathcal{F}_n$.

Монотонной элементарной конъюнкцией назовем произведение попарно различных переменных без отрицаний. *Рангом* монотонной элементарной конъюнкции назовем число ее переменных. Будем считать 1 вырожденной монотонной элементарной конъюнкцией ранга 0. Каждую булеву функцию однозначно можно записать *полиномом* вида $\sum_{i=1}^{l} X_i$, где X_i — попарно различные монотонные элементарные конъюнкции, суммирование ведется по mod 2. *Степенью* полинома называется наибольший ранг его слагаемых.

Работа выполнена при поддержке РФФИ, гранты 06-01-00438 и 07-01-00444.

Будем говорить, что функция $f(x_1, \ldots, x_n)$ *принадлежит классу* C_m, $m = 0, 1, 2, \ldots$, если она задается полиномом степени не выше m. Очевидно, что класс C_0 содержит только константы 0 и 1, класс C_1 совпадает с классом L линейных функций.

Назовем *производной* функции $f(x_1, \ldots, x_n)$ по переменной x_i функцию $f_{x_i}(x_1, \ldots, x_n)$, равную

$$f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \oplus f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n).$$

Теорема 1. *Функция* $f(x_1, \ldots, x_n)$ *принадлежит классу* C_m, $m \geqslant 1$, *тогда и только тогда, когда для каждой переменной* x_i, $i = 1, \ldots, n$, *функция* $f_{x_i}(x_1, \ldots, x_n)$ *принадлежит классу* C_{m-1}.

Доказательство. Следует из определения производной и однозначности представления булевой функции полиномом. □

Пусть булева функция $f(x_1, \ldots, x_n)$ задана вектором α_f своих значений на наборах, перечисленных в лексикографическом порядке. Число координат вектора α_f равно $N = 2^n$.

Под *алгоритмом* мы будем понимать RAM, выполняющую битовые операции; под *сложностью* алгоритма — функцию зависимости максимального числа выполненных им битовых операций от длины входных данных.

Теорема 2. *Существует алгоритм, который по вектору* α_f *со сложностью* $O(N)$ *определяет, является ли функция* $f(x_1, \ldots, x_n)$ *линейной, и в случае положительного ответа строит ее полином.*

Доказательство. Рассмотрим следующий алгоритм.

$i := 1$;
$f_i(x_i, \ldots, x_n) := f(x_1, \ldots, x_n)$;
$p(x_1, \ldots, x_n) := f(0, \ldots, 0)$.

Начало цикла.

1. Строим производную $f_{i\ x_i}(x_i, \ldots, x_n)$: делим вектор α_{f_i} пополам и суммируем покоординатно. При этом будет затрачено 2^{n-i+1} операций.
2. Если
   - $f_{x_i}(x_i, \ldots, x_n) \equiv 1$, то $p(x_1, \ldots, x_n) := p(x_1, \ldots, x_n) \oplus x_i$;
   - $f_{x_i}(x_i, \ldots, x_n) \equiv 0$, то $p(x_1, \ldots, x_n)$ оставляем без изменения;
   - иначе — алгоритм заканчивает работу и ответ «Функция нелинейна».

3. $i := i + 1$, $f_i(x_i, \ldots, x_n) = f_{i-1}(0, x_i, \ldots, x_n)$, . Для построения вектора α_{f_i} требуется 2^{n-i+1} операций.
4. Если
   - $i > n$, то алгоритм заканчивает работу, ответ «Функция линейна» и ее полином записан в $p(x_1, \ldots, x_n)$;
   - иначе — переход на начало цикла.

Корректность предложенного алгоритма следует из теоремы 1.
Подсчитаем сложность алгоритма. Она равна

$$2^n + 2^{n-1} + \ldots + 1 \leqslant 2 \cdot 2^n = O(N). \quad \square$$

Теорема 3. *Пусть задано число $m \geqslant 1$. Существует алгоритм, который по вектору α_f со сложностью $O(N)$ определяет, принадлежит ли функция $f(x_1, \ldots, x_n)$ классу C_m, и в случае положительного ответа строит ее полином.*

Доказательство. Построим требуемый алгоритм A_m индуктивно.
Базис индукции. Пусть $m = 1$. Тогда в качестве алгоритма A_1 рассмотрим алгоритм, описанный в теореме 2. Его сложность равна $2 \cdot 2^n$.
Индуктивный переход. Пусть алгоритм уже A_{m-1} построен и его сложность равна $c_{m-1} \cdot 2^n$, где c_{m-1} — некоторая константа. Рассмотрим в качестве алгоритма A_m следующий:

$i := 1$;
$f_i(x_i, \ldots, x_n) := f(x_1, \ldots, x_n)$;
$P(x_1, \ldots, x_n) := f(0, \ldots, 0)$.

Начало цикла.

1. Строим производную $f_{i\,x_i}(x_i, \ldots, x_n)$: делим вектор α_{f_i} пополам и суммируем покоординатно. При этом будет затрачено 2^{n-i+1} операций.
2. При помощи алгоритма A_{m-1} проверяем, принадлежит ли функция $f_{x_i}(x_i, \ldots, x_n)$ классу C_{m-1}.
   Если
   - ответ «да» и ее полином $p(x_1, \ldots, x_n)$, то
     $$P(x_1, \ldots, x_n) := P(x_1, \ldots, x_n) \oplus x_i \cdot p(x_1, \ldots, x_n);$$
   - иначе — алгоритм заканчивает работу и ответ «Функция не принадлежит классу C_m».

На шаге 2 мы затратим $c_{m-1} \cdot 2^{n-i+1}$ операций.

3. $i := i + 1$, $f_i(x_i, \ldots, x_n) = f_{i-1}(0, x_i, \ldots, x_n)$. Для построения вектора α_{f_i} требуется 2^{n-i+1} операций.
4. Если
   - $i > n$, то алгоритм заканчивает работу, ответ «Функция принадлежит классу C_m» и ее полином записан в $P(x_1, \ldots, x_n)$;
   - иначе — переход на начало цикла.

Корректность предложенного алгоритма следует из теоремы 1.
Подсчитаем сложность алгоритма. Она равна

$$(c_{m-1} + 1) \cdot \left(2^n + 2^{n-1} + \ldots + 1\right) \leqslant 2 \cdot (c_{m-1} + 1) \cdot 2^n = c_m \cdot 2^n,$$

где c_m — некоторая константа. Таким образом, $c_m = 2 \cdot (c_{m-1} + 1)$, $c_1 = 2$. Нетрудно подсчитать, что $c_m = 2 \cdot 2^m$. Следовательно, сложность алгоритма равна $(2 \cdot 2^m) \cdot 2^n = O(N)$, так как число m — фиксировано. □

Литература

[1] *Гаврилов Г. П.*, *Сапоженко А. А.* Задачи и упражнения по дискретной математике. М.: ФИЗМАТЛИТ, 2004.

О метриках, изометричных относительно группы сдвигов

Б. А. Погорелов, М. А. Пудовкина

1. Введение

Одной из актуальных задач в криптографии является проблема приближения функций функциями из заданного класса, в частности, аппроксимация двоичных функций линейными и мономиальными функциями. При этом естественно возникающий вопрос состоит в том, каким образом измерить расстояние между функций. Если класс функций «хорош» относительно одной меры, то будет ли он также хорош относительно другой? Обычно в качестве такой меры между булевыми функциями используется метрика Хемминга. В работе [1], посвященной аппроксимации функций над произвольным полем $\mathrm{GF}(2^t)$, в качестве основного параметра, характеризующего степень близости функции и её статистического аналога, используется функция «согласия». Эта функция отличается нормировкой, от введённой в работе [2] функции «близость». Кроме того, она не является метрикой. Представляют интерес свойства функций относительно различных метрик. В частности, исследование свойств аффинных функций и бент-функций (как функций максимально далёких от множества аффинных функций).

В данной работе исследуются метрики, группа изометрий которых содержит группу сдвигов. К таким метрикам относится и метрика Хемминга. В качестве примера показано существование метрик в этом множества, относительно которых аффинные функции находятся на разном расстоянии друг от друга. Кроме того, приведены такие метрики, что функции, максимально далекие от аффинных функций относительно метрики Хемминга, не все являются такими относительно метрик из рассматриваемого класса.

Обозначения: $\mathbb{N}_0$ — множество натуральных чисел с нулем, $n \in \mathbb{N}_0$, $n \geqslant 2$; V_t — векторное пространство t-мерных двоичных векторов; χ_t — метрика Хемминга на V_t; F_n — множество всех двоичных функций от n переменных; $\psi\colon V_n \to Z_{2^n}$ — (естественное) взаимно однозначное соот-

Работа выполнена при поддержке гранта Президента РФ НШ 4.2008.10.

ветствие между V_n и Z_{2^n}, т. е. $\psi\colon (\alpha_1, \ldots, \alpha_n) \to \sum\limits_{i=0}^{n-1} \alpha_{n-i} 2^i$; $\varepsilon_{j,t} \in V_t$, $\varepsilon_{j,t} = (0, \ldots, 0, \underset{j}{1}, 0, \ldots, 0)$, $j \in \{\overline{1, t}\}$; $\overline{a, b} = a, a+1, \ldots, b$, $a < b$; $\Delta_i^{(t)} = \{\alpha \in V_t \mid \|\alpha\| = i\}$, $i \in \{\overline{1, t}\}$; $F_n = \{f\colon V_n \to \{0, 1\}\}$; AF_n — множество всех аффинных функций из F_n; BF_n — множество всех бент-функций из F_n;

$$d_f^{\mathrm{AF}}(\mu) = \min\{\mu(a, f) \mid a \in \mathrm{AF}_n\};$$
$$d^{\mathrm{AF}}(\mu) = \max\Big\{ d_f^{\mathrm{AF}}(\mu) \Big| \ f \in F_n\Big\};$$
$$A + B = \{\alpha + \beta \mid \alpha \in A,\ \beta \in B\}, \quad A, B \subset V_n;$$
$$A + \beta = A + B, \quad B = \{\beta\},\ \beta \in V_n.$$

Если размерность t пространства V_t понятна из контекста, то у метрики χ_t, множества $\Delta_i^{(t)}$ и вектора $\varepsilon_{j,t}$ символ «t» будем опускать.

2. Свойства метрик, изометричных относительно группы сдвигов

Для описания связей между классами метрик удобны понятия подметрики и надметрики данной метрики. Отметим, что понятие подметрики метрики Хемминга предложено Б. А. Погореловым [3], а все групповые подметрики метрики Хемминга описаны в работе [4]. Пусть $|X| = n$ и μ — метрика на множестве X.

Определение 1. Метрика $\rho_\mu\colon X \times X \to \mathbb{N}_0$, удовлетворяющая для любых $\alpha, \beta, \gamma, \delta \in X$ свойствам:

1) если $\mu(\alpha, \beta) = \mu(\gamma, \delta)$, то $\rho_\mu(\alpha, \beta) = \rho_\mu(\gamma, \delta)$,

2) $\rho_\mu(\alpha, \beta) \leqslant \mu(\alpha, \beta)$,

называется подметрикой метрики μ.

Определение 2. Метрика $\rho_\mu\colon X \times X \to \mathbb{N}_0$, удовлетворяющая для любых $\alpha, \beta, \gamma, \delta \in X$ свойствам:

1) если $\rho_\mu(\alpha, \beta) = \rho_\mu(\gamma, \delta)$, то $\mu(\alpha, \beta) = \mu(\gamma, \delta)$,

2) $\mu(\alpha, \beta) \leqslant \rho_\mu(\alpha, \beta)$,

называется надметрикой метрики μ.

Очевидно, что метрика μ является подметрикой метрики ρ_μ. Обозначим через M_n^+ множество всех метрик на V_n, группа изометрий которых

содержит группу сдвигов, и имеющих вид

$$\mu(\alpha, \alpha') = \begin{cases} 0, & \text{если } \alpha = \alpha', \\ i, & \text{если } (\alpha, \alpha') \in B_i,\ i \in \overline{\{1, d\}}, \end{cases}$$

где $d \geqslant 1$, $\{B_1, \ldots, B_d\}$ — разбиение множества $V_n \times V_n \setminus \bigcup\limits_{\alpha \in V_n} (\alpha, \alpha)$.

Очевидно, что для таких метрик справедливо равенство $\mu(\alpha, \alpha + \beta) = \mu(\gamma, \gamma + \beta)$ для всех $\alpha, \gamma, \beta \in V_n$. Произвольная метрика $\mu \in M_n^+$ однозначно задаётся множествами $A_j(\mu) = \left\{\alpha \in V_n \,\middle|\, \mu(\alpha, \vec{0}) = j\right\}$, $j = 0, 1, \ldots$, а любая $(d+1)$-значная метрика $\mu \in M_n^+$ может быть представлена в виде

$$\mu(\alpha, \alpha') = \begin{cases} 0, & \text{если } \alpha + \alpha' = \vec{0}, \\ i, & \text{если } \alpha + \alpha' \in A_i(\mu),\ i \in \overline{\{1, d\}}, \end{cases}$$

где $\{A_1(\mu), \ldots, A_d(\mu)\}$ — некоторое разбиение $V_n \setminus \{\vec{0}\}$.

Назовём μ максимальной метрикой множества $M' \subset M_n^+$, если не существует у метрики μ надметрики из множества M', отличной от метрики μ.

Опишем все максимальные метрики множества M_n^+, являющиеся надметриками метрики Хемминга. Для этого приведём индуктивный способ построения некоторых метрик.

Обозначим $V_{n,i} = \left\langle \varepsilon_j \,\middle|\, j \in \overline{\{1, n\}} \setminus \{i\} \right\rangle$, $i \in \overline{\{1, n\}}$.

Лемма 1. *Пусть $n \geqslant 3$, $d \geqslant 2$, r — произвольное число из $\overline{\{1, n\}}$ и μ — произвольная $(d+1)$-значная метрика из M_{n-1}^+, β — произвольный ненулевой вектор из $V_n \setminus V_{n,r}$. Тогда функция $\rho_\mu \colon V_n \times V_n \to \mathbb{N}_0$, заданная условиями*

$$\rho_{\mu,r}^{(\beta)}(\alpha, \alpha') = \begin{cases} \mu(\alpha, \alpha'), & \text{если } \alpha + \alpha' \in V_{n,r}, \\ d + 1 + \mu(\alpha, \alpha' + \beta), & \text{если } \alpha + \alpha' \in V_{n,r} + \beta, \end{cases}$$

является $2(d+1)$-значной метрикой.

Лемма 1 позволяет описать все максимальные метрики множества M_n^+.

Для произвольных t, $t \geqslant 1$, линейно независимых векторов $\beta_1, \ldots, \beta_t$ из V_n обозначим $V_n(\beta_1, \ldots, \beta_t) = \langle \beta_1, \ldots, \beta_t \rangle$. Ясно, что $V_n(\beta_1, \ldots, \beta_t) \cong V_t$.

Утверждение 1. *При $n \geqslant 2$ каждому упорядоченному набору $(\beta_1, \ldots, \beta_n)$ линейно независимых векторов из V_n соответствует максимальная 2^n-значная метрика $\chi_{n,(\beta_1,\ldots,\beta_n)}$ множества M_n^+, заданная условиями*:

$$\chi_{n,\beta_1}(\alpha, \alpha') = \begin{cases} 0, & \text{если } \alpha + \alpha' = \vec{0}, \\ 1, & \text{если } \alpha + \alpha' = \beta_1, \end{cases}$$

если $\alpha + \alpha' \in V_n(\beta_1)$,

$$\chi_{n,\beta_1,\dots,\beta_t}(\alpha,\,\alpha') = \begin{cases} \chi_{n,\beta_1,\dots,\beta_{t-1}}(\alpha,\,\alpha'), \\ \qquad \textit{если } \alpha + \alpha' \in V_n(\beta_1,\,\dots,\,\beta_{t-1}), \\ \chi_{n,\beta_1,\dots,\beta_{t-1}}(\alpha,\,\alpha' + \beta_t) + 2^{t-1}, \\ \qquad \textit{если } \alpha + \alpha' + \beta_t \in V_n(\beta_1,\,\dots,\,\beta_{t-1}), \end{cases}$$

если $\alpha + \alpha' \in V_n(\beta_1,\,\dots,\,\beta_{t-1},\,\beta_t)$, $2 \leqslant t \leqslant n$. *Кроме того, любая максимальная метрика множества* M_n^+ *совпадает с метрикой* $\chi_{n,\beta_1,\dots,\beta_n}$ *для некоторого набора* $(\beta_1,\,\dots,\,\beta_n) \in V_n^n$, *а число максимальных метрик множества* M_n^+ *равно* $(2^n - 1)(2^n - 2)\dots\left(2^n - 2^{n-1}\right)$.

Опишем максимальные метрики множества M_n^+, являющиеся надметриками метрики Хемминга.

Следствие 1. *Пусть выполнены условия утверждения* 1, *а упорядоченный набор* $(\beta_1,\,\dots,\,\beta_n)$ *линейно независимых векторов из пространства* V_n *таков, что* $\beta_1 \in \Delta_1$ *и* $\beta_t \in \bigcup\limits_{i=1}^{2^{t-1}} \Delta_i$ *для всех* $t \in \in \left\{\overline{2,\,\lfloor \log_2 n \rfloor}\right\}$. *Тогда* $\chi_{n,\beta_1,\dots,\beta_n}$ — *максимальная* 2^n*-значная метрика множества* M_n^+, *являющаяся надметрикой метрики Хемминга. Число таких максимальных* 2^n*-значных надметрик метрики Хемминга равно*

$$\prod_{t=1}^{\lfloor \log_2 n \rfloor} \left(\sum_{i=0}^{2^{t-1}} \binom{n}{i} - 2^{t-1} \right) \cdot \prod_{j=\lfloor \log_2 n \rfloor + 1}^{n-1} (2^n - 2^j).$$

Приведём одно свойство, позволяющее сводить исследование всех максимальных метрик множества M_n^+ к одной.

Утверждение 2. *Пусть* $(\beta_1,\,\dots,\,\beta_n)$, $(\omega_1,\,\dots,\,\omega_n)$ — *два различных базиса пространства* V_n *и* $\pi \in \mathrm{GL}_n$, $\pi\colon \beta_i \to \omega_i$ *для всех* $i \in \left\{\overline{1,\,n}\right\}$. *Пусть также* $2 \leqslant d \leqslant 2^n - 1$ *и* $\mu_{\beta_1,\dots,\beta_n}$ — $(d+1)$*-значная подметрика максимальной метрики* $\chi_{n,\beta_1,\dots,\beta_n}$ *множества* M_n^+. *Тогда функция* $\mu_{\omega_1,\dots,\omega_n}\colon V_n \times V_n \to \mathbb{N}_0$, *заданная условием*

$$\mu_{\omega_1,\dots,\omega_n}(\alpha,\,\alpha') = \mu_{\beta_1,\dots,\beta_n}\left(\alpha^{\pi^{-1}},\,\alpha'^{\pi^{-1}}\right),$$

является $(d+1)$*-значной подметрикой метрики* $\chi_{n,\omega_1,\dots,\omega_n}$. *Кроме того,* $A_{j_2}(\mu_{\omega_1,\dots,\omega_n}) = (A_{j_2}(\mu_{\beta_1,\dots,\beta_n}))^\pi$, $A_{j_1}(\chi_{n,\omega_1,\dots,\omega_n}) = (A_{j_1}(\chi_{n,\beta_1,\dots,\beta_n}))^\pi$ *для всех* $j_1 \in \left\{\overline{1,\,2^n-1}\right\}$, $j_2 \in \left\{\overline{1,\,d}\right\}$.

Таким образом, каждой подметрике метрики $\chi_{n,\beta_1,\dots,\beta_n}$ однозначно соответствует подметрика метрики $\chi_{n,\omega_1,\dots,\omega_n}$. Кроме того, все максимальные метрики множества M_n^+ линейно эквивалентны. Перечислим все

$(n+1)$-значные подметрики максимальных метрик множества M_n^+ линейно эквивалентные метрики Хемминга и являющиеся подметриками надметрик метрики Хемминга.

Следствие 2. *Пусть* $\beta_1, \ldots, \beta_n$ *— базис пространства* V_n *таков, что* $\beta_1 \in \Delta_1$ *и* $\beta_t \in \bigcup\limits_{i=1}^{2^{t-1}} \Delta_i$ *для всех* $t \in \{\overline{2, \lfloor \log_2 n \rfloor}\}$. *Пусть также* $\pi \in \mathrm{GL}_n$ *и* $\pi\colon \varepsilon_i \to \beta_i$ *для всех* $i \in \{\overline{1, n}\}$. *Тогда функция* $\mu_{\beta_1,\ldots,\beta_n}\colon V_n \times V_n \to \mathbb{N}_0$, *заданная условием*

$$\mu_{\beta_1,\ldots,\beta_n}(\alpha, \alpha') = \chi(\alpha^{\pi^{-1}}, \alpha'^{\pi^{-1}}),$$

является $(n+1)$*-значной подметрикой* 2^n*-значной надметрики* $\chi_{n,\beta_1,\ldots,\beta_n}$ *метрики Хемминга.*

3. Свойства аффинных и бент-функций относительно метрик множества $M_{2^n}^+$

Приведём примеры метрик из множества $M_{2^n}^+$, относительно которых аффинные функции находятся на разном расстоянии.

Для произвольных функций $f_2, f_1 \in F_n$ обозначим через $f_2 + f_1$ функцию из F_n, заданную условием $(f_2 + f_1)(\alpha) = f_2(\alpha) + f_1(\alpha)$ для всех $\alpha \in V_n$. Обозначим также $\vec{f} = \big(f(\vec{0}), \ldots, f(\vec{1})\big)$ вектор значений функции $f \in F_n$.

Утверждение 3. *Пусть* $m = 2^n$, $r \in \{\overline{1, m}\}$, *и метрика* $\mu\colon F_n \times F_n \to \{\overline{1, m}\}$ *задана равенством*

$$\mu(f_1, f_2) = \begin{cases} \chi_m(\vec{f}_1, \vec{f}_2), & \text{если } \chi_m(\vec{f}_1, \vec{f}_2) < 2^{n-1}, \\ 2^{n-1}, & \text{если } \chi_m(\vec{f}_1, \vec{f}_2) = 2^{n-1} \\ & \quad \text{и } f_1\big(\psi^{-1}(r-1)\big) \neq f_2\big(\psi^{-1}(r-1)\big), \\ 2^{n-1}+1, & \text{если } \chi_m(\vec{f}_1, \vec{f}_2) = 2^{n-1} \\ & \quad \text{и } f_1\big(\psi^{-1}(r-1)\big) = f_2\big(\psi^{-1}(r-1)\big), \\ \chi_m(\vec{f}_1, \vec{f}_2) + 1, & \text{если } \chi_m(\vec{f}_1, \vec{f}_2) > 2^{n-1}. \end{cases}$$

Тогда расстояние между аффинными функциями $f_1, f_2 \in \mathrm{AF}_n$ *равно*

$$\mu(f_1, f_2) = \begin{cases} 2^{n-1}, & \text{если } f_1\big(\psi^{-1}(r-1)\big) \neq f_2\big(\psi^{-1}(r-1)\big), \\ 2^{n-1}+1, & \text{если } f_1\big(\psi^{-1}(r-1)\big) = f_2\big(\psi^{-1}(r-1)\big). \end{cases}$$

Утверждение 4. *Пусть* $m = 2^n$ *и* a_1, a_2 *— произвольные аффинные функции,* $\vec{a}_i \notin \{\vec{0}, \vec{1}\}$, $i \in \{1, 2\}$. *Пусть также базис* $\beta_1, \ldots, \beta_m$ *пространства* V_m *таков, что*

1) $\beta_t \in \Delta_1^{(m)}$ *для всех* $t \in \{\overline{1,\, n}\}$;

2) $\beta_c = \vec{a}_1$ *для произвольного числа* $c \in \{\overline{n+1,\, m}\}$;

3) $\|\beta_t\| \leqslant 2^m - 2$ *для всех* $t \in \{\overline{n+1,\, m}\} \setminus \{c\}$;

4) $\vec{a}_2 \notin \{\beta_1, \ldots, \beta_m\}$ *и* $\beta_i + \beta_j = \vec{a}_2$ *для некоторых* $i, j \in \{\overline{1,\, m}\}$;

Пусть также $\pi \in \mathrm{GL}_n$, $\pi\colon \varepsilon_i \to \beta_i$ *для всех* $i \in \{\overline{1,\, m}\}$, *и метрика* $\mu\colon F_n \times F_n \to \{\overline{1,\, m}\}$ *задана равенством*

$$\mu_{\beta_1,\ldots,\beta_m}(f_1,\, f_2) = \chi_m\bigl(\vec{f}_1^{\,\pi^{-1}},\, \vec{f}_2^{\,\pi^{-1}}\bigr).$$

Тогда существуют аффинные функции $f_1,\, f_2 \in \mathrm{AF}_n$, *для которых*

$$\mu_{\beta_1,\ldots,\beta_m}(f_1,\, f_1 + a_1) = \mu_{\beta_1,\ldots,\beta_m}(f_2,\, f_2 + a_1) = 1,$$
$$\mu_{\beta_1,\ldots,\beta_m}(f_1,\, f_1 + a_2) = \mu_{\beta_1,\ldots,\beta_m}(f_2,\, f_2 + a_2) = 2,$$
$$\mu_{\beta_1,\ldots,\beta_m}(f_1,\, f_2) > 3.$$

Приведём некоторые (2^n+1)-значные подметрики максимальных метрик множества $M_{2^n}^+$, которые «разделяют» множество бент-функций.

Пусть $H_i(f) = \{\vec{a} + \vec{f}\,|\, a \in \mathrm{AF}_n,\ \chi(\vec{f},\, \vec{a}) = i\}$ и $B_\gamma^{(1)} = \{f \in \mathrm{BF}_n \mid \gamma \in H_r(f)\}$, $B_\gamma^{(2)} = \{f \in \mathrm{BF}_n \mid \gamma \notin H_r(f)\}$, где $\gamma \in V_{2^n}$, $r = 2^{n-1} - 2^{n/2-1}$.

Утверждение 5. *Пусть*

1) *n чётно,* $m = 2^n$, $t \in \{\overline{n,\, m}\}$, $r = 2^{n-1} - 2^{n/2-1}$;

2) $(\varepsilon_{i_1}, \ldots, \varepsilon_{i_t}, \gamma, \varepsilon_{i_{t+1}}, \ldots, \varepsilon_{i_{m-1}})$ *— произвольный базис пространства* V_m, *где* $\gamma \in \Delta_r^{(m)}$;

3) $\pi \in \mathrm{GL}_m$ *и* $\pi\colon \varepsilon_{i_l} \to \varepsilon_{i_l}$ *для всех* $l \in \{\overline{1,\, m-1}\}$, $\pi\colon \varepsilon_{i_m} \to \gamma$, *где* $\{i_j \,|\, j \in \{\overline{1,\, m}\}\} = \{\overline{1,\, m}\}$;

4) $\mu(\alpha,\, \alpha') = \chi\bigl(\alpha^{\pi^{-1}},\, \alpha'^{\pi^{-1}}\bigr)$ *для всех* $(\alpha,\, \alpha') \in V_n \times V_n$.

Тогда $d_{f_1}^{\mathrm{AF}}(\mu) = 1$ *и* $d_{f_1}^{\mathrm{AF}}(\mu) < d_{f_2}^{\mathrm{AF}}(\mu)$ *для любых функций* $f_1 \in B_\gamma^{(1)}$, $f_2 \in B_\gamma^{(2)}$.

Литература

[1] *Кузьмин А. С.*, *Нечаев А. А.*, *Шишкин В. А.* Бент- и гипербент-функции над конечным полем // Труды по дискретной математике. 2007. Т. 10. С. 97–122.

[2] *Солодовников В. И.* Бент-функции из конечной абелевой группы // Дискретая математика. 2002. Т. 14, № 1. С. 99–113.

[3] *Погорелов Б. А.* Подметрики метрики Хемминга и теорема А. А. Маркова // Труды по дискретной математике. 2006. Т. 9.

[4] *Погорелов Б. А.*, *Пудовкина М. А.* Подметрики Хемминга и их группы изометрий // Труды по дискретной математике. 2008. Т. 11.

О некоторых свойствах совершенно уравновешенных булевых функций

С. В. Смышляев

1. Основные определения и обозначения

Пусть $\mathbb{F}_2$ — поле Галуа из двух элементов, $\mathrm{V}_n = \mathbb{F}_2^n$ — пространство наборов длины $n \in \mathbb{N}$ над полем $\mathbb{F}_2$. Будем обозначать через $\mathcal{F}_n$ — множество булевых функций от n переменных $\{x_1, x_2, \ldots, x_n\}$. Крайними переменными функции $f \in \mathcal{F}_n$ будем называть x_1 и x_n. Через Φ_n будем обозначать подмножество функций из $\mathcal{F}_n$, существенно зависящих от обеих крайних переменных. Пусть $m \in \mathbb{N}$. Рассмотрим систему булевых уравнений:

$$f(x_s, x_{s+1}, \ldots, x_{s+n-1}) = y_s, \quad s = 1, 2, \ldots, m. \tag{1}$$

Обозначим для $f \in \mathcal{F}_n$ через f^* отображение из V_{m+n-1} в V_m вида:

$$f^*(x_1, x_2, \ldots, x_{m+n-1}) = \\ = (f(x_1, \ldots, x_n), f(x_2, \ldots, x_{n+1}), \ldots, f(x_m, \ldots, x_{m+n-1})). \tag{2}$$

Определение 1 ([1]). Булева функция $f \in \mathcal{F}_n$ называется совершенно уравновешенной, если соотношение

$$\#(f^*)^{-1}(y) = 2^{n-1}$$

выполняется для любого $m \in \mathbb{N}$ и любого $y \in \mathrm{V}_m$. Множество совершенно уравновешенных функций из $\mathcal{F}_n$ обозначим $\mathcal{PB}_n$.

Замечание 1. Легко видеть (см. [1]), что если функция линейна хотя бы по одной из крайних переменных, то она является совершенно уравновешенной. Обозначим множество функций из $\mathcal{F}_n$, линейных по первой переменной, $\mathcal{L}_n$, а множество функций из $\mathcal{F}_n$, линейных по последней переменной, $\mathcal{R}_n$.

2. Предварительные результаты

Отметим преобразования множества $\mathcal{F}_n$, оставляющие инвариантным множество $\mathcal{PB}_n$ (см. [1]):

1) $\gamma_0 \colon f(x_1, \ldots, x_n) \to f(x_1, \ldots, x_n) \oplus 1$;
2) $\gamma_1 \colon f(x_1, \ldots, x_n) \to f(x_1 \oplus 1, \ldots, x_n \oplus 1)$;
3) $\gamma_2 \colon f(x_1, \ldots, x_n) \to f(x_n, \ldots, x_1)$.

Теорема 1 ([2]). *Пусть $n \in \mathbb{N}$ и $f \in \mathcal{F}_n$. Пусть*

$$\{X_m = (x_1, \ldots, x_{m+n-1})\}_{m=1}^{\infty}$$

— последовательность случайных векторов с распределением

$$\Pr\{X_m = (a_1, \ldots, a_{m+n-1})\} = 2^{-(m+n-1)}$$

для любых $(a_1, \ldots, a_{m+n-1}) \in \mathrm{V}_{m+n-1}$. Случайный вектор $Y_m = f^*(X_m)$ распределен равномерно для любого $m \in \mathbb{N}$ тогда и только тогда, когда f — совершенно уравновешенная функция.

Теорема 2 ([1]). *Булева функция $f \in \mathcal{F}_n$ является совершенно уравновешенной тогда и только тогда, когда не существует двух двоичных последовательностей*

$$x = (x_1, \ldots, x_r),\ z = (z_1, \ldots, z_r) \in \mathrm{V}_r,\ r > 2n,$$

таких, что

$$x_1 = z_1, \ldots, x_n = z_n,\ x_{r-n+1} = z_{r-n+1}, \ldots, x_r = z_r; \tag{3}$$

$$x \neq z; \tag{4}$$

$$f(x_i, \ldots, x_{i+n-1}) = f(z_i, \ldots, z_{i+n-1}), \quad i = 1, \ldots, r-n+1. \tag{5}$$

Для пары натуральных чисел m, k рассмотрим отображение $\Xi_{m,k}$ из $\mathcal{F}_m \times \mathcal{F}_k$ в $\mathcal{F}_{m+k-1}$ вида

$$\Xi_{m,k}(g, h) = g[h] = f \in \mathcal{F}_{m+k-1}, \quad g \in \mathcal{F}_m,\ h \in \mathcal{F}_k, \tag{6}$$

где

$$f(x_1, \ldots, x_{m+k-1}) = g[h](x_1, \ldots, x_{m+k-1}) = \\ = g(h(x_1, \ldots, x_k), h(x_2, \ldots, x_{k+1}), \ldots, h(x_m, \ldots, x_{m+k-1})).$$

Для этой конструкции справедливо следующее утверждение.

Теорема 3 ([3]). *Пусть $g \in \mathcal{F}_m$, $h \in \mathcal{F}_k$. Функция $f = g[h] \in \mathcal{F}_{m+k-1}$ совершенно уравновешена тогда и только тогда, когда функции g и h совершенно уравновешены.*

Утверждение теоремы 3 позволяет строить булевы функции из $\mathcal{PB}_n$, не входящие в классы $\mathcal{L}_n$ и $\mathcal{R}_n$.

Для любой $f \in \Phi_n$ обозначим $f_\gamma(x_1, \ldots, x_N) \equiv f(x_{N-\gamma_n}, x_{N-\gamma_{n-1}}, \ldots, x_{N-\gamma_1})$, где $\gamma = (\gamma_1, \ldots, \gamma_n)$ — набор неотрицательных целых чисел, такой, что $\gamma_1 = 0$, $\forall i \in \{1, \ldots, n-1\}\ \gamma_{i+1} > \gamma_i$, $N = \gamma_n + 1$. Далее будем рассматривать только такие наборы γ, определяющие точки входа в фильтрующем генераторе.

В работе [4] Голичем было сформулировано и доказано (в одну сторону) следующее утверждение о совершенно уравновешенных булевых функциях:

Теорема 4 ([4]). *Пусть на вход кодирующего устройства с функцией f_γ поступает случайная последовательность, биты которой независимы и принимают значения* 0 *и* 1 *с вероятностями* 1/2. *Тогда выходная последовательность кодирующего устройства обладает тем же свойством при любом выборе набора γ тогда, когда f линейна по крайней существенной переменной.*

Утверждение теоремы 4 является тривиальным следствием теоремы 1 и замечания 1. В работе [4] было также высказано предположение, что утверждение теоремы 4 верно и в обратную сторону («и только тогда»), однако доказательства предложено не было. Воспользуемся аппаратом совершенно уравновешенных функций, переформулируем и докажем предположенное утверждение.

3. Основные результаты

Теорема 5. *Для любой функции $\Phi_n \setminus (\mathcal{L}_n \cup \mathcal{R}_n)$ существует набор γ такой, что $f_\gamma \notin \mathcal{PB}_N$.*

Доказательство. Здесь приведем доказательство, предполагающее, что f не зависит ни от одной переменной линейно.

Выберем набор γ следующим образом: $\gamma_1 = 0, \forall i \in \{1, \ldots, n-1\}\ \gamma_{i+1} > 2\gamma_i$, и докажем разрешимость следующей системы:

$$
\begin{cases}
f(x_{N-\gamma_n-1}, x_{N-\gamma_{n-1}-1}, \ldots, x_{N-\gamma_2-1}, x_{N-\gamma_1-1}) = \\
\qquad = f(z_{N-\gamma_n-1}, z_{N-\gamma_{n-1}-1}, \ldots, z_{N-\gamma_2-1}, z_{N-\gamma_1-1}), \\
f(x_{N-\gamma_n}, x_{N-\gamma_{n-1}}, \ldots, x_{N-\gamma_2}, x_{N-\gamma_1}) = \\
\qquad = f(z_{N-\gamma_n}, z_{N-\gamma_{n-1}}, \ldots, z_{N-\gamma_2}, z_{N-\gamma_1}), \\
f(x_{N-\gamma_n+1}, x_{N-\gamma_{n-1}+1}, \ldots, x_{N-\gamma_2+1}, x_{N-\gamma_1+1}) = \\
\qquad = f(z_{N-\gamma_n+1}, z_{N-\gamma_{n-1}+1}, \ldots, z_{N-\gamma_2+1}, z_{N-\gamma_1+1}), \\
\ldots \\
f(x_{N-\gamma_1}, x_{N-\gamma_1+\gamma_n-\gamma_{n-1}}, \ldots, x_{N-2\gamma_1+\gamma_n}) = \\
\qquad = f(z_{N-\gamma_1}, z_{N-\gamma_1+\gamma_n-\gamma_{n-1}}, \ldots, z_{N-2\gamma_1+\gamma_n}), \\
f(x_{N-\gamma_1+1}, x_{N-\gamma_1+\gamma_n-\gamma_{n-1}+1}, \ldots, x_{N-2\gamma_1+\gamma_n+1}) = \\
\qquad = f(z_{N-\gamma_1+1}, z_{N-\gamma_1+\gamma_n-\gamma_{n-1}+1}, \ldots, z_{N-2\gamma_1+\gamma_n+1}), \\
x_{N-\gamma_n-1} = z_{N-\gamma_n-1}, \quad x_{N-\gamma_n} = z_{N-\gamma_n}, \\
\ldots \\
x_{N-\gamma_1-1} = z_{N-\gamma_1-1}, \quad x_{N-\gamma_1} = 0, \quad z_{N-\gamma_1} = 1, \\
x_{N-\gamma_1+1} = z_{N-\gamma_1+1}, \quad x_{N-\gamma_1+2} = z_{N-\gamma_1+2}, \\
\ldots \\
x_{N-2\gamma_1+\gamma_n+1} = z_{N-2\gamma_1+\gamma_n+1}.
\end{cases}
\tag{7}
$$

Все равенства, в которые не входит явным образом $x_{N-\gamma_1}$ и $z_{N-\gamma_1}$ выполняются автоматически, остается разрешить следующую систему:

$$\begin{cases} f(x_{N-\gamma_n+\gamma_1}, x_{N-\gamma_{n-1}+\gamma_1}, \ldots, x_{N-\gamma_2+\gamma_1}, 0) = \\ \qquad = f(x_{N-\gamma_n+\gamma_1}, x_{N-\gamma_{n-1}+\gamma_1}, \ldots, x_{N-\gamma_2+\gamma_1}, 1), \\ f(x_{N-\gamma_n+\gamma_2}, x_{N-\gamma_{n-1}+\gamma_2}, \ldots, x_{N-\gamma_3+\gamma_2}, 0, x_{N-\gamma_1+\gamma_2}) = \\ \qquad = f(x_{N-\gamma_n+\gamma_2}, x_{N-\gamma_{n-1}+\gamma_2}, \ldots, x_{N-\gamma_3+\gamma_2}, 1, x_{N-\gamma_1+\gamma_2}), \\ \ldots \\ f(0, x_{N-\gamma_{n-1}+\gamma_n}, \ldots, x_{N-\gamma_1+\gamma_n}) = \\ \qquad = f(1, x_{N-\gamma_{n-1}+\gamma_n}, \ldots, x_{N-\gamma_1+\gamma_n}). \end{cases} \tag{8}$$

Каждое из уравнений этой системы имеет решение ввиду того, что f не зависит линейно ни от одного из своих аргументов. Покажем, что любые два уравнения этой системы — j-е и l-е — независимы. Требуется показать:

$$\forall\, i, j, k, l \in \{1, \ldots, n\}\ i \neq j,\ k \neq l,\ i \neq k,\ j \neq l \implies \\ \implies N - \gamma_i + \gamma_j \neq N - \gamma_k + \gamma_l. \tag{9}$$

Очевидно, достаточно показать, что $\forall\, i \in \{1, \ldots, n-1\}\ \forall\, j \leqslant i\ \forall\, k \leqslant i\ \forall\, l\ \gamma_{i+1} - \gamma_j > \gamma_k - \gamma_l$. Мы выбирали набор γ из условия $\forall\, i \in \{1, \ldots, n-1\}$ $\gamma_{i+1} > 2\gamma_i$, поэтому $\gamma_{i+1} - \gamma_j > \gamma_i + (\gamma_i - \gamma_j) \geqslant \gamma_i \geqslant \gamma_i - \gamma_l \geqslant \gamma_k - \gamma_l$.

Таким образом, верны неравенства (9), а значит, система (8) состоит из n независимых уравнений, каждое из которых разрешимо. Поэтому системы (8) и, соответственно, (7) совместны, что, по теореме 2, означает $f_\gamma \notin \mathcal{PB}_N$, что и требовалось доказать. □

Здесь приведено доказательство для случая функции, не зависящей линейно ни от одного аргумента. Для общего случая доказательство принципиально такое же — мы выбираем набор γ определенного вида, а затем по теореме 2 доказываем отсутствие совершенной уравновешенности f_γ — но оно намного больше по объему из-за необходимости особым образом учитывать линейные переменные функции f.

Введем понятие барьера булевой функции, тесно связанное со свойством совершенной уравновешенности.

Определение 2 ([5]). Булева функция $f \in \mathcal{F}_n$ называется функцией с правым барьером длины b, если система уравнений

$$\begin{cases} f(y_1, y_2, \ldots, y_n) = f(z_1, z_2, \ldots, z_n), \\ f(y_2, y_3, \ldots, y_{n+1}) = f(z_2, z_3, \ldots, z_{n+1}), \\ \ldots \\ f(y_{b-1}, y_b, \ldots, y_{b+n-2}) = f(z_{b-1}, z_b, \ldots, z_{b+n-2}), \\ y_1 = z_1 = x_1, \ldots, y_{n-1} = z_{n-1} = x_{n-1},\ y_n = 0,\ z_n = 1 \end{cases} \tag{10}$$

имеет решение, а система

$$\begin{cases} f(y_1, y_2, \ldots, y_n) = f(z_1, z_2, \ldots, z_n), \\ f(y_2, y_3, \ldots, y_{n+1}) = f(z_2, z_3, \ldots, z_{n+1}), \\ \ldots \\ f(y_{b-1}, y_b, \ldots, y_{b+n-2}) = f(z_{b-1}, z_b, \ldots, z_{b+n-2}), \\ f(y_b, y_{b+1}, \ldots, y_{b+n-1}) = f(z_b, z_{b+1}, \ldots, z_{b+n-1}), \\ y_1 = z_1 = x_1, \ldots, y_{n-1} = z_{n-1} = x_{n-1}, \ y_n = 0, \ z_n = 1 \end{cases} \tag{11}$$

решений не имеет.

Булева функция $f \in \mathcal{F}_n$ называется функцией с левым барьером длины b, если $f^{\gamma_2}(x_1, \ldots, x_n) \equiv f(x_n, \ldots, x_1)$ является функцией с правым барьером длины b.

Булева функция $f \in \mathcal{F}_n$ имеет барьер, если она имеет правый или левый барьер, или оба сразу. При этом длиной барьера функции называется соответственно длина правого барьера, левого барьера, или меньшая из длин барьеров.

Теорема 6. *Наличие барьера у функции является достаточным, но не необходимым условием совершенной уравновешенности функции.*

Доказательство. Если у функции есть правый или левый барьер, то непосредственно из определения барьера, а также теоремы 2 вытекает совершенная уравновешенность функции.

Покажем, что существуют совершенно уравновешенные функции без барьера. Для этого рассмотрим функцию

$$f(x_1, x_2, x_3, x_4, x_5) = x_1 \oplus x_2 \oplus x_5 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1x_4 \oplus \\ \oplus x_2x_3 \oplus x_2x_4 \oplus x_3x_4 \oplus x_4x_5 \oplus x_1x_3x_4 \oplus x_2x_3x_4 = g^{\gamma_2}[g],$$

где $g(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3 \oplus x_2x_3$, $g^{\gamma_2}(x_1, x_2, x_3) \equiv g(x_3, x_2, x_1)$. Функция g имеет левый барьер длины $k = 1$, следовательно, g, $g^{\gamma_2} \in \mathcal{PB}_3$. Используя утверждение теоремы 3, получим: $f \in \mathcal{PB}_5$. Чтобы доказать отсутствие барьеров у функции f, рассмотрим две пары последовательностей:

$$\begin{bmatrix} 0,0,0,1,(0,1,0,1,0,0,)(0,1,0,1,0,0,)(0,1,0,1,0,0,)\ldots \\ 0,0,0,1,(1,0,0,0,1,0,)(1,0,0,0,1,0,)(1,0,0,0,1,0,)\ldots \end{bmatrix}, \tag{12}$$

$$\begin{bmatrix} \ldots(0,0,1,1,1,1,1,0,)(0,0,1,1,1,1,1,0,)(0,0,1,1,1,1,1,0,)0,0,0,1,0,0 \\ \ldots(1,1,1,0,0,0,1,1,)(1,1,1,0,0,0,1,1,)(1,1,1,0,0,0,1,1,)0,1,0,1,0,0 \end{bmatrix}. \tag{13}$$

Для любого наперед заданного $k > 0$ продолжим пару (12) до длины большей $k + 3$, воспользовавшись ее периодичностью. Затем, подставив

её в системы уравнений (10) и (11), убеждаемся, что обе системы совместны, то есть, никакое k не удовлетворяет определению 2, — следовательно, у функции f нет правого барьера. Аналогично, воспользовавшись парой (13), убедимся в отсутствии у f левого барьера. Следовательно, $f \in \mathcal{PB}_5$ является функцией без барьера. □

Анализируя системы (10) и (11) из определения барьера, можно доказать следующее утверждение, позволяющее классифицировать функции с правым (левым) барьером:

Теорема 7 ([5]). *Пусть $f \in \mathcal{F}_n$ — функция с правым (левым) барьером длины b, $b < n$, а $g \in \mathcal{F}_{n-b}$ — произвольная функция. Тогда функция $h(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \oplus g(x_1, \ldots, x_{n-b})$ (соответственно, $h(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \oplus g(x_{b+1} \ldots, x_n)$) тоже является функцией с правым (левым) барьером длины b.*

Длина барьера — величина, определенным образом характеризующая обратимость отображения f^*, порожденного совершенно уравновешенной функцией f. Из общих соображений понятно, что в множестве $\mathcal{PB}_n$ наибольший интерес представляют функции с большой длиной барьера, а также функции без барьера.

При доказательстве теоремы 6 мы доказали, что взятая нами композиция функций $g^{\gamma_2}[g]$ является функцией без барьера. Можно показать, что g не имеет правого барьера, а g^{γ_2} не имеет левого барьера. Возникает вопрос: а всегда ли предложенный способ композиции функций $g[h]$ сохраняет наличие и отсутствие правых и левых барьеров? Ответ на этот вопрос дает следующая теорема.

Теорема 8. *Пусть $h \in \mathcal{F}_k$, $g \in \mathcal{F}_m$, $f = g[h]$, а длины правых (левых) барьеров функций h, g, f равны, соответственно b, c, d. Тогда выполнено соотношение*
$\max\{b, c\} \leqslant d \leqslant b + c - 1$, где оба неравенства могут обращаться, а могут не обращаться в равенства.[1]

Обозначим основные идеи доказательства. Неравенства $d \geqslant b$ и $d \leqslant b + c - 1$ как для случая конечных b, c, d, так и в случае, если какие-то из функций не имеют барьеров, доказываются с помощью анализа систем, которые получаются из (10), (11) подстановкой вместо аргументов функции g значений функции h на соответствующих наборах. По полученным системам и определяются неравенства относительно b и $b + c - 1$, накладываемые на d. Чтобы доказать неравенство $d \geqslant c$, мы пользуемся

[1] Здесь мы формально считаем функцию без правого (левого) барьера функцией с правым (левым) барьером равным $+\infty$.

леммой 1, доказательство которой опирается только на определение совершенно уравновешенной функции.

Лемма 1. *Если* $f \in \mathcal{PB}_n$, *то для любого натурального* u, *для любых наборов* $z_0, z_1 \in \mathrm{V}_u$ *существуют натуральное число* r *и наборы* $x, y \in \mathrm{V}_{r+n-1}$, $z \in \mathrm{V}_{r-u}$ *такие, что выполнена система*

$$\begin{cases} x_1 = y_1, \\ \dots \\ x_{n-1} = y_{n-1}, \\ f^*(x) = (z|z_0), \\ f^*(y) = (z|z_1) \end{cases}$$

Для завершения доказательства теоремы 8 рассматриваются функции $f_1(x_1, x_2, x_3, x_4) = x_3 \oplus x_2x_4(x_1 \oplus 1)$ и $f_2(x_1, x_2, x_3, x_4) = x_3 \oplus x_1x_4(x_2 \oplus 1)$ с правыми барьерами длины $b = c = 3$. Несложно проверить непосредственно, что у функций $f_2[f_2]$, $f_1[f_1]$ и $f_1[f_2]$ длины правых барьеров равны $d = 3, d = 4, d = 5$ соответственно. Во всех случаях $\max\{b, c\} = 3$, $b + c - 1 = 5$, то есть d принимает все три возможных значения между $\max\{b, c\}$ и $b + c - 1$, что и завершает доказательство теоремы.

Приведем пример способа построения совершенно уравновешенных функций без правого барьера:

Лемма 2. *Функции вида*

$$\begin{aligned} f = {} & x_1 \oplus x_{m_1^\circ} x_{m_1^\circ+1} \dots x_{m_1} h_1(x_{m_1+1}, x_{m_1+2}, \dots, x_{m_k}) \\ & \oplus x_{m_2^\circ} x_{m_2^\circ+1} \dots x_{m_2} h_2(x_{m_2+1}, x_{m_2+2}, \dots, x_{m_k}) \oplus \dots \\ & \oplus x_{m_k^\circ} x_{m_k^\circ+1} \dots x_{m_k}, \quad 1 < m_1^\circ < m_1 < m_2^\circ < m_2 < \dots < m_k^\circ < m_k, \end{aligned} \tag{14}$$

где h_i *— элементарные мономы, а* k *— нечетное, являются совершенно уравновешенными функциями без правого барьера.*

Доказательство. Все такие функции линейны по первой переменной, поэтому, как следует из замечания 1, они являются совершенно уравновешенными. Докажем, что у них отсутствует правый барьер. Возьмем произвольную функцию f указанного вида. Для любого сколь угодно большого b рассмотрим следующую пару последовательностей:

$$\begin{array}{l} \underbrace{0, 0, \dots, 0,}_{m_k-1} \underbrace{1, 0, 1, \dots, 0,}_{m_k^\circ - m_{k-1}^\circ} \underbrace{0, 0, 0, \dots, 0,}_{m_{k-1}^\circ - m_{k-2}^\circ} \underbrace{1, 0, 1, \dots, 1,}_{m_{k-2}^\circ - m_{k-3}^\circ} \quad \dots \\ \dots \quad \underbrace{0, 1, 0, \dots, 0,}_{m_3^\circ - m_2^\circ} \underbrace{0, 0, 0, \dots, 0,}_{m_2^\circ - m_1^\circ} \underbrace{1, 0, 1, \dots, 0,}_{m_1^\circ - 1} \underbrace{1, 1, \dots, 1}_{b - m_k - m_k^\circ + 2} \\ \underbrace{0, 0, \dots, 0,}_{m_k-1} \underbrace{0, 1, 0, \dots, 1,}_{m_k^\circ - m_{k-1}^\circ} \underbrace{0, 0, 0, \dots, 0,}_{m_{k-1}^\circ - m_{k-2}^\circ} \underbrace{1, 0, 1, \dots, 1,}_{m_{k-2}^\circ - m_{k-3}^\circ} \quad \dots \\ \dots \quad \underbrace{1, 0, 1, \dots, 1,}_{m_3^\circ - m_2^\circ} \underbrace{0, 0, 0, \dots, 0,}_{m_2^\circ - m_1^\circ} \underbrace{0, 1, 0, \dots, 1,}_{m_1^\circ - 1} \underbrace{0, 0, \dots, 0}_{b - m_k - m_k^\circ + 2} \end{array}$$

Подставляя эту пару последовательностей в систему из определения 2, получим, что у f нет правого барьера длины b или меньше. Это верно для сколь угодно большого b, поэтому у f нет правого барьера. □

Теперь с помощью леммы 2 и теоремы 8 мы можем описать крупный класс совершенно уравновешенных функций без барьера.

Теорема 9. *Пусть* f_1, f_2 *имеют вид* (14) *или получены из функций вида* (14) *с помощью преобразования*

$$\gamma_1 : f(x_1, \ldots, x_n) \to f(x_1 \oplus 1, \ldots, x_n \oplus 1).$$

Тогда функции вида $f_1[f_2^{\gamma_2}]$ *и* $f_1^{\gamma_2}[f_2]$ *являются совершенно уравновешенными функциями без барьера.*

Функции, полученные с помощью теоремы 9, обладают всеми положительными криптографическими качествами совершенно уравновешенных булевых функций, в частности, фильтрующие генераторы с такими функциями устойчивы к оптимальной корреляционной атаке Андерсона [6]. При этом у них отсутствует нежелательное для фильтрующей функции свойство барьера конечной длины, которое позволяет эффективно находить прообраз f^*. Поэтому к фильтрующим генераторам с такими функциями неприменима инверсионная атака Голича [4], как и всякая другая атака, использующая наличие у фильтрующей функции барьера конечной длины.

Литература

[1] *Сумароков С. Н.* Запреты двоичных функций и обратимость для одного класса кодирующих устройств // Обозрение прикладной и промышленной математики. 1994. Т. 1, вып. 1. С. 33–55.

[2] *Логачёв О. А.*, *Сальников А. А.*, *Ященко В. В.* Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2004.

[3] *Логачёв О. А.* Об одном классе совершенно уравновешенных булевых функций // Материалы Третьей международной научной коференции по проблемам безопасности и противодействия терроризму. Москва, МГУ имени М. В. Ломоносова (25–27 октября 2007 г.). М.: МЦНМО, 2008. С. 137–141.

[4] *Golić, J. Dj.* On the Security of Nonlinear Filter Generators / D. Gollmann (ed.) // Proc. of Fast Software Encryption 1996. (LNCS, v. 1039.) P. 173–188. Springer, 1996.

[5] *Логачёв О. А.*, *Ященко В. В.*, *Смышляев С. В.* Новые методы изучения совершенно уравновешенных булевых функций (в печати).

[6] *Anderson R. J.* Searching for the Optimum Correlation Attack / B. Preneel (ed.) // Proc. Fast Software Encryption 1995. (LNCS, v. 1008.) P. 137–143. Springer, 1995.

Оптимальность выбора функции XOR в одной модели дифференциального криптоанализа хэш-функций семейства MDx

Г. А. Карпунин, Нгуен Т. Х.

1. Семейство хэш-функций MDx

В 1990 году Р. Ривест разработал хэш-функцию MD4 [1], на основе которой возникло целое семейство хэш-функций MDx. К этому семейству можно отнести такие хэш-функции, как MD5, HAVAL, RIPEMD, SHA и некоторые другие. Все хэш-функции из семейства MDx относятся к хэш-функциям, сконструированным «с нуля» (dedicated hash functions), и имеют итерационную структуру. На t-ой итерации таких хэш-функций вычисляется промежуточное хэш-значение $H_t = h_c(H_{t-1}, M_t)$ с помощью функции сжатия h_c, зависящей от предыдущего хэш-значения H_{t-1} и текущего блока M_t исходного сообщения. В свою очередь, вычисление функции сжатия h_c часто также имеет итерационную структуру. Для MD4 вычисление $\widetilde{H} = \mathrm{MD4}_c(H, M)$ выглядит следующим образом:

$$\begin{cases} (w_1, \ldots, w_{48}) = \mathrm{MessageExpansion}(M), \\ (Q_{-3}, Q_0, Q_{-1}, Q_{-2}) = H, \\ Q_j = (f_j(Q_{j-1}, Q_{j-2}, Q_{j-3}) + Q_{j-4} + w_j + c_j) \lll s_j, \quad j = 1, \ldots, 48, \\ \widetilde{H} = (Q_{-3} + Q_{45}, Q_0 + Q_{48}, Q_{-1} + Q_{47}, Q_{-2} + Q_{46}), \end{cases} \tag{1}$$

где w_j, Q_j — вспомогательные 32-битные переменные; MessageExpansion — операция расширения 512-битного блока M до 1536 бит; f_j — фиксированные булевы функции, действующие на свои аргументы побитово; $+$ — сложение по mod 2^{32}; c_j, s_j — фиксированные аддитивные константы и константы сдвига; $\lll$ — операция циклического сдвига в сторону старших бит.

В работе [1] не приводится обоснование выбора фиксированных значений для f_j, c_j и s_j. Насколько нам известно такого обоснования в открытой печати к настоящему моменту не появилось. В данной работе делается попытка оценить влияние выбора булевых функций f_j на стойкость

хэш-функции MD4 (а также конструктивно близких к ней) к методам дифференциального криптоанализа.

Методы дифференциального криптоанализа являются одними из наиболее эффективных методов нахождения коллизий у хэш-функций, сконструированных «с нуля». В 2004–2005 годах группа китайских криптоаналитиков под руководством С. Вонг [2, 3, 4, 5, 6] успешно применила эти методы к широко распространенным хэш-функциям семейства MDx — MD4, MD5, HAVAL-128, RIPEMD, SHA-0, SHA-1. Для первых пяти хэш-функций были найдены реальные коллизии, а для SHA-1 понижена стойкость по сравнению с общими методами на основе парадокса задачи о днях рождения.

2. Дифференциальный криптоанализ. Общий подход

Рассмотрим некоторую функцию $Y = \varphi(X)$, вычисление которой может быть задано с помощью последовательности уравнений с использованием вспомогательной переменной Z. Запишем уравнение (в общем случае это может быть системой уравнений), в котором вычисление функции φ связывает X, Y, Z:

$$\Phi(X, Y, Z) = 0. \tag{2}$$

Пусть значения переменных X, Y, Z принадлежат некоторым конечным коммутативным группам $(G_x, +_x)$, $(G_y, +_y)$, $(G_z, +_z)$ соответственно.

Под *дифференциальным криптоанализом функции* φ мы будем понимать следующую последовательность действий противника.

- Противник выбирает согласно своей стратегии некоторые значения $\Delta X \in G_x$, $\Delta Y \in G_y$, $\Delta Z \in G_z$, называемые *входной*, *выходной* и *промежуточной разностью* соответственно. Совокупность этих разностей $\xi = (\Delta X, \Delta Y, \Delta Z)$ назовем *дифференциальной характеристикой*.
- Противник добавляет к уравнению (2) разностное уравнение

  $$\Phi(X + \Delta X, Y + \Delta Y, Z + \Delta Z) = 0$$

  и рассматривает разностную систему

  $$\begin{cases} \Phi(X, Y, Z) = 0 \\ \Phi(X + \Delta X, Y + \Delta Y, Z + \Delta Z) = 0. \end{cases} \tag{3}$$

  Вероятность P_ξ решения этой системы уравнений относительно X, Y, Z назовем *вероятностью дифференциальной характеристики* ξ.

От классического определения вероятности дифференциальной характеристики наше определение отличается постоянным множителем. Поскольку по заданному входному значению X выходное значение Y и значение вспомогательной переменной Z определяется однозначно, то $P_\xi^{\text{кл.}} = |G_y| \cdot |G_z| \cdot P_\xi$.

- Противник пытается подобрать дифференциальную характеристику $\xi \neq 0$ таким образом, чтобы максимизировать ее вероятность P_ξ.
- После выбора дифференциальной характеристики ξ противник пытается угадать решение разностной системы (3), равновероятно выбирая его из множества всех допустимых значений $G_x \times G_y \times G_z$. Вероятность успеха однократной попытки угадывания решения, очевидно, равна P_ξ.

Канонической стратегией действий противника мы будем называть произвольное подмножество $\Xi \subset G_x \times G_y \times G_z$ множества всех возможных дифференциальных характеристик ξ. На практике обычно коммутативные группы G_x, G_y, G_z представляют собой векторные пространства вида $(g_1, g_2, \ldots, g_R) \in \left(\mathbb{F}_2^T\right)^R$. В таком случае среди канонических стратегий выделим *типичные стратегии* Ξ, которые задаются фиксированием некоторых компонент $g_i = 0$, а остальные нефиксированные компоненты принимают произвольные значения.

В качестве показателя эффективности действий противника по одной из канонических стратегий Ξ мы примем среднее $\mathrm{EP}(\Xi) = (1/|\Xi|) \sum_{\xi \in \Xi} P_\xi$. Чем меньше среднее $\mathrm{EP}(\Xi)$, тем менее эффективны действия противника.

3. Дифференциальный криптоанализ. Хэш-функция MD4

Применим общий подход из раздела 2 к хэш-функции MD4. Для упрощения анализа исключим циклические сдвиги в системе (1). Тогда, переобозначив переменные и введя дополнительные, полученную систему можно переписать в следующем достаточно общем виде:

$$\begin{cases} F(X_{jk}, A_{jk}, Z_i) = 0 \\ f_j(X_{j1}, \ldots, X_{jn}) = f_j(X_{j1} + A_{j1}, \ldots, X_{jn} + A_{jn}) + A_{j0}, \end{cases} \tag{4}$$

где X_{jk}, Z_i — это T-битные переменные; $A_{jk} (= \Delta X_{jk})$ — T-битные параметры системы; $+$ — сложение по $\bmod 2^T$.

Выделим одно уравнение из системы (4), опустив индекс j:

$$f(X_1, \ldots, X_n) = f(X_1 + A_1, \ldots, X_n + A_n) + A_0. \tag{5}$$

Лемма 1. *Пусть* Ξ *— каноническая стратегия. Предположим, что решения системы* (4) $\setminus$ (5) (*система* (4) *без уравнения* (5)) *распределены равномерно на стратегии* Ξ; *и решения системы* (4) $\setminus$ (5) *и уравнения* (5) *независимы на* Ξ. *Тогда*

$$\mathrm{EP}(\Xi) = \frac{1}{2}\,\mathrm{EP}_f(\Xi).$$

Эта лемма позволяет нам оценить влияние каждого конкретного уравнения (5) на показатель эффективности $\mathrm{EP}(\Xi)$ действий противника по типичной стратегии Ξ.

Назовем среднее значение $\mathrm{EP}_f(\Xi)$ *уступчивостью* булевой функции f дифференциальному криптоанализу по канонической стратегии Ξ. Определим *характеристический вектор уступчивости* $\chi_T(f)$ булевой функции f к типичным стратегиям как совокупность средних значений $\mathrm{EP}_f(\Xi)$ по всем типичным стратегиям Ξ, исключая дублирование:

$$\chi_T(f) = \Big(\mathrm{EP}_f^T(0,\,0,\,\ldots,\,\underset{i_1}{*},\,\ldots,\,\underset{i_k}{*},\,\ldots,\,0)\Big)_{\{i_1,\ldots,i_k\}\subset\{1,\ldots,n+1\}},$$

где

$$\mathrm{EP}_f^T(0,\,0,\,\ldots,\,\underset{i_1}{*},\,\ldots,\,\underset{i_k}{*},\,\ldots,\,0) = \\ = \frac{1}{2^k}\sum_{(A_{i_1},\ldots,A_{i_k})\in\mathbb{F}_2^{kT}} P_f^T(0,\,0,\,\ldots,\,A_{i_1},\,\ldots,\,A_{i_k},\,\ldots,\,0),$$

а $P_f^T(A_0,\,A_1,\,\ldots,\,A_n)$ — вероятность решения уравнения (5) при фиксированных значениях параметров $A_0,\,A_1,\,\ldots,\,A_n$.

Будем считать *булеву функцию* f_2 *более уступчивой к дифференциальному криптоанализу, чем* f_1, если и только если

$$\chi_T(f_1) \prec \chi_T(f_2),$$

где $\preceq$ — естественное отношение частичного порядка на векторах из $\mathbb{R}^N$: $(v_1,\,\ldots,\,v_N) \preceq (u_1,\,\ldots,\,u_N) \iff v_i \leqslant u_i$ для всех $i = 1,\,\ldots,\,N$.

Сформулированный критерий оценки уступчивости вводит на множестве всех булевых функций от n переменных $\mathcal{F}_n$ отношение частичного порядка $(\mathcal{F}_n,\,\preceq_T)$, которое, вообще говоря, зависит от T. Для этого отношения в данной работе находится наименьший элемент, который является наименее уступчивой (наиболее устойчивой) булевой функцией к дифференциальному криптоанализу хэш-функций семейства MDx.

Для формулировки теоремы, позволяющей вычислять вектор уступчивости $\chi_T(f)$, введем несколько определений. Пусть A и B — два числа. Будем говорить, что перенос в $(T+1)$-й бит суммы $A + B$ равен 1, если эта сумма, рассматриваемая как целочисленная, больше или равна 2^T, и равен 0

в противном случае. Через $P^T_{g;\alpha_0\alpha_1\ldots\alpha_n}(A_0, A_1, \ldots, A_n)$, $(\alpha_0\alpha_1\ldots\alpha_n) \in \mathbb{F}_2^{n+1}$, мы обозначим вероятностную меру тех решений уравнения (5), для которых переносы в $(T+1)$-й бит выражений $f(X_1 + A_1, X_2 + A_2, \ldots, X_n + A_n) + A_0$, $X_1 + A_1, \ldots, X_n + A_n$ равны $\alpha_0, \alpha_1, \ldots, \alpha_n$ соответственно. Обозначим через $M_{a_0a_1\ldots a_n}$ квадратную матрицу размера $2^{n+1} \times 2^{n+1}$, состоящую из элементов $P^1_{f;\beta_0\beta_1\ldots\beta_n}(a_0 + \alpha_0, a_1 + \alpha_1, \ldots, a_n + \alpha_n)$. Строчки этой матрицы заиндексированы двоичными векторами $(\beta_0\beta_1\ldots\beta_n)$, а столбцы — двоичными векторами $(\alpha_0\alpha_1\ldots\alpha_n)$.

Теорема 1. *Имеет место равенство*

$$\mathrm{EP}^T_f(0, 0, \ldots, \underset{i_1}{*}, \ldots, \underset{i_k}{*}, \ldots, 0) = \\ = (11\ldots1) \cdot \left(\frac{1}{2^k} \sum_{(a_{i_1}\ldots a_{i_k}) \in \mathbb{F}_2^k} M_{00\ldots a_{i_1}\ldots a_{i_k}\ldots0} \right)^T \cdot \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

4. Случай $n = 3$, $T = 32$

Теорема 1 позволяет вычислить отношение уступчивости $(\mathcal{F}_n, \preceq_T)$ для наиболее важного случая $n = 3$, $T = 32$, имеющего место для многих хэш-функций семейства MDx, в том числе MD4. На рис. 1 в виде ориентированного графа показано отношение $(\mathcal{F}_3, \preceq_{32})$, ограниченное на функции, существенно зависящие от всех своих переменных. В рамках находятся булевы функции, выбранные Р. Ривестом для MD4. Из рис. 1 видно, что функция $\mathrm{XOR} = x_1 \oplus x_2 \oplus x_3$ является наименьшим элементом для отношения $(\mathcal{F}_3, \preceq_{32})$. Оказывается, что такая же ситуация имеет место и в случае произвольных n, T.

5. Случай произвольных n, T

Теорема 2. *Для любого* $T \geqslant 1$ *отношение частичного порядка* $(\mathcal{F}_n, \preceq_T)$ *совпадает с* $(\mathcal{F}_n, \preceq_1)$.

Теорема 3. 1. *Для функции* $\mathrm{XOR} = x_1 \oplus \ldots \oplus x_n$

$$\chi_1(\mathrm{XOR}) = \left[\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}\right].$$

2. *Для любой функции* $f \in \mathcal{F}_n$

$$\chi_1(\mathrm{XOR}) \preceq \chi_1(f).$$

Следствие 1. *Для отношения частичного порядка* $(\mathcal{F}_n, \preceq_T)$ *существует наименьший элемент, и этим элементом является функция* XOR.

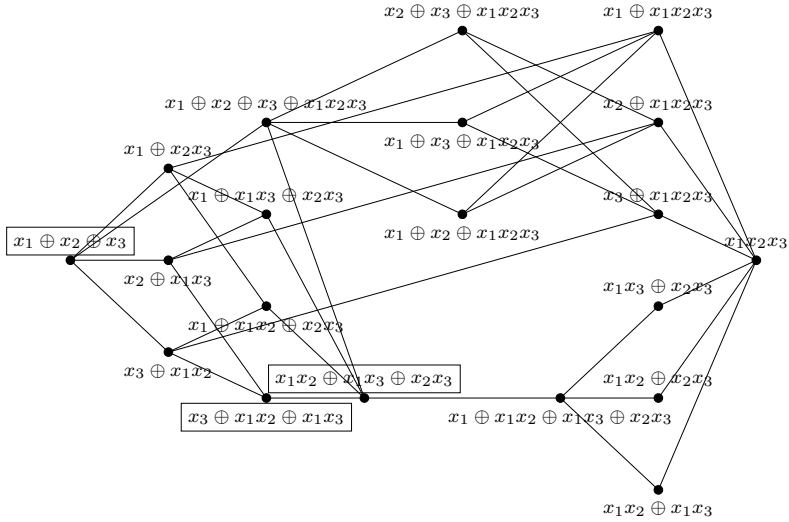


Рис. 1. Отношение уступчивости ($\mathcal{F}_3$, $\preceq_{32}$)

Таким образом показано, что в рассмотренной модели дифференциального криптоанализа хэш-функций семейства MDx наименее уступчивой (наиболее устойчивой) является функция XOR.

Литература

[1] *Rivest R.* The MD4 message digest algorithm // Proceedings of CRYPTO'90, LNCS 537. Springer-Verlag, 1991, p. 303–311.

[2] *Wang X.*, *Feng D.*, *Lai X.*, *Yu H.* Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD // IACR ePrint archive (`http://eprint.iacr.org/`), report 2004/199.

[3] *Wang X.*, *Lai X.*, *Feng D.*, *Chen H.*, *Yu X.* Cryptanalysis of the Hash Functions MD4 and RIPEMD // Proceedings of EUROCRYPT'2005, LNCS 3494. Springer-Verlag, 2005, p. 1–18.

[4] *Wang X.*, *Yu H.*. How to Break MD5 and Other Hash Functions // Proceedings of EUROCRYPT'2005, LNCS 3494. Springer-Verlag, 2005, p. 19–35.

[5] *Wang X.*, *Yu H.*, *Yin Y. L.* Efficient Collision Search Attacks on SHA-0 // Proceedings of CRYPTO'2005, LNCS 3621. Springer-Verlag, 2005, p. 1–16.

[6] *Wang X.*, *Yu H.*, *Yin Y. L.* Finding Collisions in the Full SHA-1 // Proceedings of CRYPTO'2005, LNCS 3621. Springer-Verlag, 2005, p. 17–36.

Комбинаторно-геометрический метод исследования взаимосвязей между шифрами

С. С. Коновалова, С. С. Титов

Работа посвящена применению комбинаторно-геометрического метода построения эндоморфных совершенных шифров [1], сформулированного авторами в [2], их современных аналогов ($U(L)$- и $O(L)$-стойкие шифры [3]), и исследования связей между ними. В работах [2, 4] была доказана теорема о взаимосвязи между линейными эндоморфными $O(2)$- и $U(2)$-стойкими шифрами. В [5] — теорема о взаимосвязи между циклическими эндоморфными $O(2)$- и $U(2)$-стойкими шифрами, а также выдвинута гипотеза о существовании такой взаимосвязи для произвольной функции зашифрования. В данной статье эта гипотеза доказывается для систем Веблена—Веддерберна (VW-систем). Получен ряд результатов по таким частным случаям системы Веблена—Веддерберна, задающим конечную плоскость, как системы Холла, почти-поля, не сводящиеся к полям, а также по группам Матьё. Доказано утверждение, необходимое для исследования и построения $O(3)$-стойких шифров.

Напомним, что эндоморфные $U(L)$- и $O(L)$-стойкие шифры — это шифры, стойкие к таким активным атакам злоумышленника, как имитация и подмена сообщения. Их построение эквивалентно построению таблицы зашифрования определенного вида, состоящей из подстановок на множестве элементов открытого текста. Если подстановки, составляющие $O(L)$-стойкий шифр, образуют группу, то она является L-транзитивной (для минимального количества ключей — точно L-транзитивной). Для случаев, которые мы рассматриваем в нашей работе, — $L = 2$ и $L = 3$ — группа подстановок является точно 2- и 3-транзитивной соответственно. Для выполнения свойства $U(L)$-стойкости необходима единственность ключа для множества из L элементов, порядок которых не важен.

В наших работах мы рассматриваем только комбинаторную часть проблемы построения современных аналогов совершенных шифров, отвлекаясь от вероятностной, связанной с генерированием случайной равновероятной гаммы, и не требует, чтобы подстановки шифра образовывали группу.

Рассмотрим вопрос взаимосвязи между $O(2)$- и $U(2)$-стойкими шифрами для случая, когда $O(2)$-стойкий шифр строится в системе Вебле-

на—Веддерберна, что является возможным благодаря наблюдению 2 из [6] о том, что построение эндоморфного $O(2)$-стойкого шифра с минимальным числом ключей сводится к построению конечной (аффинной) плоскости. Проективная плоскость здесь не рассматривается в контексте этой проблемы. Оформим наблюдение в виде теоремы, проверив для $O(2)$-стойкого шифра свойства конечной плоскости [7].

Первое свойство — через две точки на плоскости проходит единственная прямая — соответствует единственности ключа в $O(2)$-стойком шифре (любые два разных x из множества открытых текстов X должны однозначно соответствовать паре y). Второе свойство — любые две непараллельные прямые пересекаются в единственной точке — доказывается аналогично, так как в противном случае прямые пересекаются в двух точках, чего не должно быть. Третье свойство — через точку, не лежащую на прямой, проходит единственная прямая, параллельная данной, — доказывается следующим образом.

Пусть (x_0, y_0) — точка на плоскости, $y = f(x)$ — функция зашифрования на некотором ключе, $f(x_0) \neq y_0$. Пусть $f(a) = y_0$, $a \in X$, $|X| = \lambda$. Тогда $a = x_m \neq x_0$ (см. рис. 1).

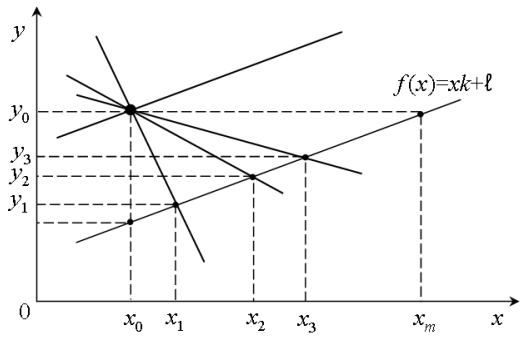


Рис. 1. Доказательство третьего свойства конечной плоскости для $O(2)$-стойкого шифра

В силу свойства $O(2)$-стойкости для любого $x_j \in X \setminus \{x_0, a\}$, $j = 1, 2, \ldots, \lambda - 2$, существует единственный ключ k_j такой, что $f_{k_j}(x_0) = y_0$, $f_{k_j}(x_j) = f(x_j)$. Поскольку множество ключей k таких, что $f_k(x_0) = y_0$, в $O(2)$-стойком эндоморфном шифре с минимальным числом ключей, имеет мощность $(\lambda - 1)$ [1], остается единственный ключ $k = k_{\lambda-1}$, для которого $f_{k_{\lambda-1}}(x_0) = y_0$. При этом кривая $y = f_{k-1}(x)$ не может иметь с исход-

ной кривой $y = f(x)$ общую точку по свойству $O(2)$-стойкости, так как все возможные абсциссы таких точек заняты ключами k_j ($j = 1, 2, \ldots, \lambda - 2$). Отсутствие точек пересечения геометрически интерпретируется как параллельность. Поэтому делаем вывод, что существует единственная кривая, проходящая через точку (x_0, y_0) и параллельная исходной кривой. Таким образом доказана

Теорема 1. *Построение эндоморфного $O(2)$-стойкого шифра с минимальным числом ключей равносильно построению конечной аффинной плоскости.*

Свойства конечной плоскости, а значит и $O(2)$-стойкого шифра, показаны на рис. 2.

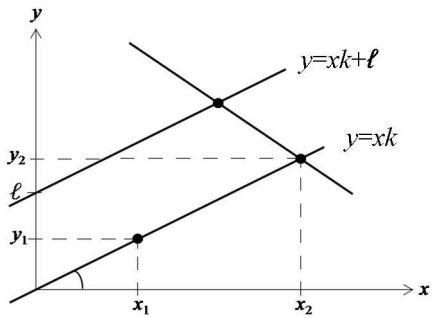


Рис. 2. Аффинная плоскость

Отметим, что геометрически $U(2)$-стойкий шифр не должен содержать такие функции зашифрования, которые показаны на обоих графиках на рис. 3, а $O(2)$-стойкий шифр — только функции для ситуации, показанной слева на этом же рисунке.

Пусть дан $U(2)$-стойкий шифр; имея в виду построение в дальнейшем системы Веблена—Веддерберна, по аналогии с циклическими массивами и линейными шифрами [2] определим уравнение зашифрования как

$$y = f_{k,\ell}(x) = xk + \ell, \quad x, \ell \in X, \ k \in K,$$

где K — некоторое множество автоморфизмов некоторой абелевой группы $(X, +)$, $0 \in X$ ($|K| = (\lambda - 1)/2$), для которых выполняется, стало быть, свойство односторонней дистрибутивности $(a + b)k = ak + bk$, причем $a \neq \neq 0 \ \& \ k \neq 0 \implies ak \neq 0$, $k \in K$. Ключом является пара (k, ℓ).

Дополним его до $O(2)$-стойкого шифра, для этого докажем ряд теорем.

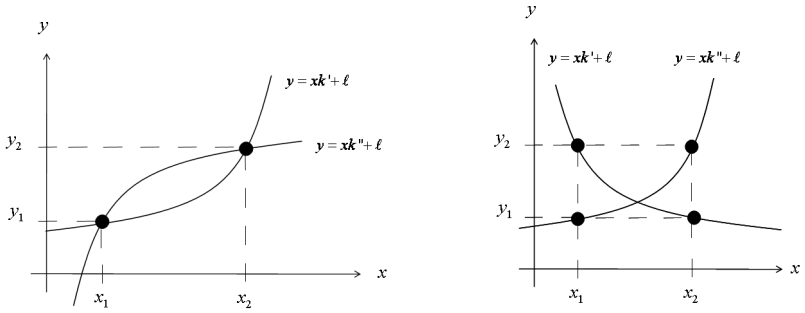


Рис. 3. Функции зашифрования, не входящие в $U(2)$-стойкий шифр

Теорема 2. *Условие $U(2)$-стойкости шифра имеет место тогда и только тогда, когда для любых k', k'' и $x \neq 0$, $k' \neq k''$ выполняются неравенства $xk' - xk'' \neq 0$ и $xk' + xk'' \neq 0$.*

Доказательство. Рассмотрим ситуацию, когда данные условия не выполняются, тогда возможны два случая.

1. Если $b = ak' = ak''$, то для обоих автоморфизмов множество $\{0,\ a\}$ зашифровывается во множество $\{0,\ b\}$, что не удовлетворяет условию $U(2)$-стойкости (а также и $O(2)$-стойкости).
2. Если $b = ak' = -ak''$, то рассмотрим две функции: $f_{k',0}(x) = xk' + 0 = xk'$ и $f_{k'',b}(x) = xk'' + b$. Очевидно, что для первой функции $\{0,\ a\} \to \{0,\ b\}$. Для второй справедливо $f_{k'',b}(0) = b$ и $f_{k'',b}(a) = ak'' + b = -b + b = 0$, то есть $f_{k'',b}(x)\colon \{0,\ a\} \to \{0,\ b\}$ тоже. Вывод аналогичен как в первом случае: условие $U(2)$-стойкости не выполняется.

Рассмотрим обратное рассуждение, когда нам дано нарушение условия $U(2)$-стойкости:

$$\begin{cases} f_{k',\ell'}\colon \{a,\ b\} \to \{c,\ d\}, \\ f_{k'',\ell''}\colon \{a,\ b\} \to \{c,\ d\}. \end{cases}$$

Возможны также два случая.

В первом случае (нарушающем и условие $O(2)$-стойкости) при

$$\begin{cases} f_{k',\ell'}(a) = f_{k'',\ell''}(a) = c, \\ f_{k',\ell'}(b) = f_{k'',\ell''}(b) = d \end{cases}$$

(аналогично ситуации на левом графике, рис. 3) имеем систему

$$\begin{cases} ak' + \ell' = ak'' + \ell'' = c, \\ bk' + \ell' = bk'' + \ell'' = d. \end{cases}$$

Применяя групповое свойство операции сложения при вычитании строк, получим $ak' - bk' = ak'' - bk'' = c - d$, откуда следует тождество $(a-b)k' = (a-b)k''$ в силу односторонней дистрибутивности (так как k', k'' — автоморфизмы). При $a \neq b$ имеем $x_0 = a - b \neq 0$, получаем $x_0 k' - x_0 k'' = 0$, противоречащее первому условию теоремы.

Во втором случае при

$$\begin{cases} f_{k',\ell'}(a) = f_{k'',\ell''}(b) = c, \\ f_{k',\ell'}(b) = f_{k'',\ell''}(a) = d \end{cases}$$

(аналогично ситуации на правом графике, рис. 3) имеем

$$\begin{cases} ak' + \ell' = bk'' + \ell'' = c, \\ bk' + \ell' = ak'' + \ell'' = d. \end{cases}$$

Снова вычитаем строки: $ak' - bk' = bk'' - ak'' = c - d$ и в силу левой дистрибутивности и ассоциативности сложения получим $(a-b)k' = (b-a)k'' = -(a-b)k''$. При $a \neq b$ имеем $x_0 = a - b \neq 0$, получаем $x_0 k' + x_0 k'' = 0$, противоречащее второму условию теоремы. Теорема доказана. □

Так как при доказательстве теоремы 1 рассматривался случай, имеющий место в том числе и для $O(2)$-стойкого шифра, то для VW-систем справедливо следующее

Следствие. *Условие $O(2)$-стойкости шифра имеет место тогда и только тогда, когда для любых k', k'' и $x \neq 0$, $k' \neq k''$ выполняется неравенство $xk' - xk'' \neq 0$.*

Таким образом, если для функции зашифрования выполняется необходимое условие $U(2)$-стойкости, то и выполняется условие $O(2)$-стойкости, но ключей при этом в два раза меньше необходимого их числа.

Лемма 1. *В VW-системе функция $f(x) = xk' - xk''$ есть биекция при $k' \neq k''$.*

Доказательство. Если $ak' - ak'' = bk' - bk''$, то $(a-b)k' = (a-b)k''$, откуда $k' = k''$ при $a - b \neq 0$ ввиду возможности левого деления. □

Определим множество K как множество ненулевых ключей, задающих $U(2)$-стойкий шифр. При этом оно является подмножеством множества

ненулевых ключей VW^*, необходимых для построения $O(2)$-стойкого шифра в VW-системе, то есть $K \subset VW^*$. Теперь сформулируем следующую лемму.

Лемма 2. *Если $K \subset VW^*$, K задает $U(2)$-стойкий шифр, то для любых k', $k'' \in K$ функция $f(x) = xk' + xk''$ является биекцией.*

Доказательство. Если $ak' + ak'' = bk' + bk''$, то в силу дистрибутивности $(a-b)k' = -(a-b)k''$. Применяя замену $c = a - b \neq 0$ получим равенство $ck' + ck'' = 0$ вопреки условию $U(2)$-стойкости по теореме 1. □

При $x = 1$ получим условие $k' \neq \pm k''$.

Будем дополнять $U(2)$-стойкий шифр до $O(2)$-стойкого по аналогии с тем, как это делалось для линейных функций в [2].

Для $O(2)$-стойкого шифра необходимо выполнение только одного из условий $U(2)$-стойкого шифра (а именно $xk' - xk'' \neq 0$). Пусть даны функции вида $f_{k,\ell}(x) = xk + \ell$, входящие в $U(2)$-стойкий шифр, $k \in K$. Рассмотрим для каждого из ℓ функции вида $y = x\overline{k} + \ell = -xk + \ell$, $\forall\, x : xk + x\overline{k} = 0$; исходя из следствия теоремы 1 они могут быть включены в $O(2)$-стойкий вместе с функциями $y = xk + \ell$. Графики этих функций можно рассматривать как прямые наклона $\overline{k}$, противоположного наклону k. Этот наклон $\overline{k}$ определяется через произведение на любой элемент $x \in X$ формулой $x\overline{k} = -xk$, где k — исходный наклон из $U(2)$-стойкого шифра. Взяв здесь $x = 1$, получим инволюцию $\overline{k} = -k$. Проверим для такого набора функций свойство $O(2)$-стойкости. Пусть такие функции не удовлетворяют этому условию, тогда существуют такие две точки x_1, x_2 $(x_1 \neq x_2)$, что

$$\begin{cases} x_1k + \ell = y_1, \\ x_1\overline{k} + \ell = y_1, \\ x_2k + \ell = y_2, \\ x_2\overline{k} + \ell = y_2 \end{cases} \Longrightarrow \begin{cases} x_1k + \ell = -x_1k + \ell = y_1, \\ x_2k + \ell = -x_2k + \ell = y_2 \end{cases} \Longrightarrow$$

$$\Longrightarrow \begin{cases} x_1k + x_1k = 0, \\ x_2k + x_2k = 0 \end{cases} \Longrightarrow 2x_1k = 2x_2k = 0,$$

откуда следует либо $x_1 = x_2 = 0$, либо $k = 0$, что противоречит начальным условиям.

Итак, пусть дан $U(2)$-стойкий шифр, тогда согласно теореме 1 выполняется неравенство $xk' \pm xk'' \neq 0$, k', $k'' \in K$. Теперь включим такие функции в $O(2)$-стойкий шифр и проверим условие из следствия: $xk' - xk'' = \pm$ $\pm(xk' \pm xk'')$, откуда получим $xk' - xk'' \neq 0$. Таким образом, для построения $O(2)$-стойкого шифра к каждой функции $y = xk + \ell$ из $U(2)$-стойкого шифра можно добавить функции вида $y = -xk + \ell = x\overline{k} + \ell = x(-k) + \ell$,

то есть расширить множество «наклонов» прямых до $K \cup \overline{K}$, при этом $K \cap \cap \overline{K} \neq 0$ по леммам 1 и 2.

Теперь рассмотрим ситуацию, когда дан $O(2)$-стойкий шифр, $y = f_{k,\ell}(x) = xk + \ell$ и необходимо выделить из него $U(2)$-стойкий шифр, в котором функции зашифрования должны удовлетворять условию, что для любого ненулевого a выполняются неравенства $ak' \neq ak'' \neq 0$ и $ak' = -ak''$.

Лемма 3. *Если в VW-системе выполняется равенство* $ak' + ak'' = 0$, $a \neq 0$, $k' \neq 0$, $k'' \neq 0$, *тогда в эндоморфный* $U(2)$*-стойкий подшифр с минимальным числом ключей будет входить либо функция* $y = xk' + \ell'$, *либо функция* $y = xk'' + \ell''$.

Доказательство. Рассмотрим две произвольные прямые вида $y = xk' + \ell'$ и $y = xk'' + \ell''$ (см. рис. 4), причем существует такая абсцисса a, что $ak' + ak'' = 0$. Имеем $k' \neq k''$, так как иначе было бы $2ak' = 0$, что при $a \neq 0$, $k' \neq 0$ противоречит нечетности λ.

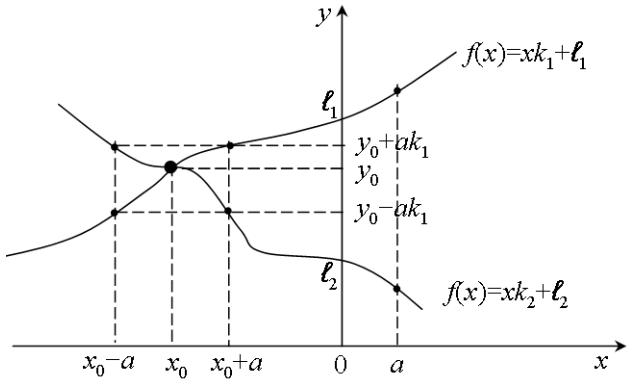


Рис. 4. Доказательство леммы 3

Пусть (x_0, y_0) — точка их пересечения:

$$x_0k' + \ell' = x_0k'' + \ell'' = y_0 \implies x_0k' - x_0k'' = \ell'' - \ell'.$$

В силу леммы 1 такая точка существует. Найдем значения обеих функций при $x = x_0 \pm a$, преобразуем их в соответствии с имеющимися тождества-

ми:

$$\begin{aligned}
f_{k',\ell'}(x_0 - a) &= (x_0 - a)k' + \ell' = x_0k' - ak' + \ell' = \\
&= (x_0k' + \ell') - ak' = y_0 - ak'; \\
f_{k'',\ell''}(x_0 + a) &= (x_0 + a)k'' + \ell'' = x_0k'' + ak'' + \ell'' = \\
&= (x_0k'' + \ell'') + ak'' = y_0 - ak'; \\
f_{k'',\ell''}(x_0 - a) &= (x_0 - a)k'' + \ell'' = x_0k'' - ak'' + \ell'' = \\
&= (x_0k'' + \ell'') - ak'' = y_0 + ak'; \\
f_{k',\ell'}(x_0 + a) &= (x_0 + a)k' + \ell' = x_0k' + ak' + \ell' = \\
&= (x_0k' + \ell') + ak' = y_0 + ak'.
\end{aligned}$$

Откуда получаем, что

$$f_{k',\ell'}(x_0 - a) = f_{k'',\ell''}(x_0 + a), \quad f_{k',\ell'}(x_0 + a) = f_{k'',\ell''}(x_0 - a),$$

то есть для обеих функций справедливо

$$\{x_0 - a,\, x_0 + a\} \mapsto \{y_0 - ak',\, y_0 + ak'\},$$

что не удовлетворяет условию $U(2)$-стойкости.

При $\ell' = \ell'' = \ell$ получаем $y_0 = \ell$, $x_0k' - x_0k'' = 0 \implies x_0 = 0$ (так как $k' \neq k''$), $\{-a,\, a\} \mapsto \{ak' + \ell,\, -ak' + \ell\}$.

При $\ell' = \ell'' = 0$ получаем $y_0 = x_0 = 0$, $\{-a,\, a\} \mapsto \{ak',\, -ak'\}$.

Отсюда делаем следующие выводы: при $ak' + ak'' = 0$ для некоторого a в $U(2)$-стойкий шифр нельзя включать для любого ℓ обе прямые $y = xk' + \ell$, $y = xk'' + \ell$, или наоборот не включать обе прямые такого вида. Также нельзя для одного из ℓ' брать $y = xk' + \ell'$, а для другого ℓ'' — функцию $y = xk'' + \ell''$. Остается единственный вариант выделения $U(2)$-стойкого шифра из $O(2)$-стойкого — это брать только либо функции вида $y = xk' + \ell'$, либо $y = xk'' + \ell''$. □

Теперь для каждого $m \in M$, где M — множество наклонов $U(2)$-стойкого шифра, и для каждого $a \in X$ определим преобразование $m \mapsto m_a$ как $m_a = a \setminus (-am)$. В частности, $m_{-1} = -m$. Тогда

$$\begin{aligned}
(m_a)_a &= a \setminus (-a(a \setminus (-am))) = \\
&= a \setminus -(a \setminus a(a \setminus (-am))) = a \setminus (-(-am)) = a \setminus (am) = m,
\end{aligned}$$

то есть это — инволюция на множестве K всех наклонов, $K = M \cup (-M)$, $(-M) = \{k : k = -m,\ m \in M\}$. Так что в VW-системе, как в $O(2)$-стойком шифре, можно выделить $U(2)$-стойкий шифр тогда и только тогда, когда эти инволюции удовлетворяют следующему условию: при $m \in M$ имеем $(((m_{a_1})_{a_2})_{\ldots})_{a_s} \in M$ тогда и только тогда, когда s четно. При $s = 1$ получаем $m_a = -m_a$ и функция $(-m_a)$ не входит в $U(2)$-стойкий шифр. При

$s = 2$ получаем $a_1 m \mapsto -a_1 m_{a_1}$, $a_2 m_{a_1} \mapsto -a_2 (m_{a_1})_{a_2}$. Таким образом, если функция входит в $U(2)$-стойкий шифр, то функция, полученная одним преобразованием — нет.

Рассмотрим рис. 5.

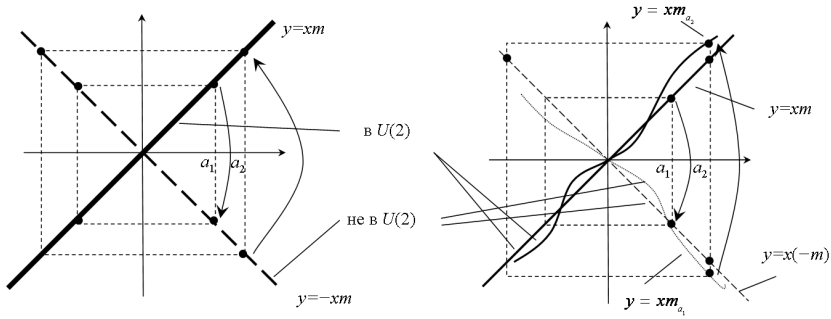


Рис. 5. Выделение $U(2)$-стойких шифров из $O(2)$-стойких

На левом графике показан пример единственной инволюции, а на правом — пример инволюций для $s = 2$.

Таким образом, доказана

Теорема 3. *Если имеется эндоморфный $U(2)$-стойкий шифр, то существует конечная плоскость, соответствующая $O(2)$-стойкому шифру, которая допускает инволюцию $m \mapsto -m$. Обратно: если имеется $O(2)$-стойкий эндоморфный шифр, представляющий собой набор прямых в системе Веблена—Веддерберна, то в нем можно выделить $U(2)$-стойкий подшифр только тогда, когда квазигруппа по умножению допускает такой набор инволюций m_a, что при $m \in M$ имеем $(((m_{a_1})_{a_2})_{\dots})_{a_s} \in M$ в том и только том случае, когда s четно.*

Итак, полное изучение взаимосвязи $U(2)$- и $O(2)$-стойких шифров приводит к изучению групп, порожденных инволюциями.

Следствиями теоремы являются утверждения, в которых накладываются ограничения на функции зашифрования массивов, в том числе циклических, претендующих на удовлетворение условиям $O(2)$- и $U(2)$-стойкости, простейшем случае, когда инволюция единственна.

Утверждение 1. *Если для двух функций зашифрования циклического массива $\alpha_{k'}(x - \ell)$ и $\alpha_{k''}(x - \ell)$ $(x, \ell \in Z_\lambda)$ выполняются равенства $\alpha_{k'}(x_1) = \alpha_{k''}(x_3)$, $\alpha_{k'}(x_2) = \alpha_{k''}(x_4)$, причем $x_3 - x_1 = x_4 - x_2$, то такие функции не могут быть включены в массивы $CA_1(2, \lambda, \lambda)$ и $CPA_1(2, \lambda, \lambda)$.*

Утверждение 2. *Если для двух функций зашифрования циклического массива* $\alpha_{k'}(x-\ell)$ *и* $\alpha_{k''}(x-\ell)$ $(x, \ell \in Z_\lambda)$ *выполняются равенства* $\alpha_{k'}(x_1)=\alpha_{k''}(x_4)$, $\alpha_{k'}(x_2)=\alpha_{k''}(x_3)$, *причем* $x_1+x_4=x_2+x_3$, *то такие функции не могут быть включены в массив* $CPA_1(2, \lambda, \lambda)$.

Утверждение 3. *Если в массиве есть функции* $\beta_{k'}(x)$ *и* $\beta_{k''}(x)$ *такие, что* $(\beta_{k'}(x)-\beta_{k''}(x)) \bmod \lambda \neq \gamma(x) \in S_\lambda$, *то этот массив не является массивом* $A_1(2, \lambda, \lambda)$ *или* $PA_1(2, \lambda, \lambda)$.

Утверждение 4. *Если в массиве есть функции* $\beta_{k'}(x)$ *и* $\beta_{k''}(x)$ *такие, что* $(\beta_{k'}(x)+\beta_{k''}(x)) \bmod \lambda \neq \gamma(x) \in S_\lambda$, *то этот массив не является массивом* $PA_1(2, \lambda, \lambda)$.

Заметим, что утверждения 3 и 4 имеют сходство с наблюдением 3 из [2, 4] о требованиях к матрицам линейного $U(2)$-стойкого шифра. Таким образом, при анализе любых массивов, претендующих на соответствие $O(2)$- или $U(2)$-стойким шифрам, с уравнением зашифрования $y = \alpha_k(x-\ell)$ или $y=\beta_k(x)+\ell$, можно проверять не сам массив на стойкость, а только его таблицу умножения, которая значительно меньше массива.

Рассмотрим пример для теоремы 3, когда $O(2)$-стойкий шифр строится в системе Холла [8] по формуле зашифрования $y=xM_k+\ell$, где M_k соответствует матрице зашифрования Холла с характеристическим многочленом $f(x)=x^2-rx-s$, неприводимым в поле $\mathrm{GF}(q)$, а элементы x, k, y, ℓ — двумерные вектора над полем $\mathrm{GF}(q)$. Напомним, что система Холла — частный случай VW-системы, задающей конечную плоскость; также она является решением [2] третьей задачи теории линейных совершенных шифров из [1] как пример линейного, но не билинейного шифра.

Обозначим за P множество матриц Холла M_k, необходимых для зашифрования. Рассмотрим вопрос выделения $U(2)$-стойких шифров из $O(2)$-стойких шифров, построенных по системе Холла. Для этого при $r=0$ достаточно доказать, что $M_{-k}=-M_k$. Действительно матрица

$$M_{-k}=\begin{pmatrix} -k_0 & -k_1 \\ \dfrac{-(-k_0)^2+s}{-k_1} & -(-k_0) \end{pmatrix}=\begin{pmatrix} -k_0 & -k_1 \\ \dfrac{k_0^2-s}{k_1} & k_0 \end{pmatrix}$$

равна матрице

$$-M_k=\begin{pmatrix} -k_0 & -k_1 \\ -\dfrac{-k_0^2+s}{k_1} & -(-k_0) \end{pmatrix}=\begin{pmatrix} -k_0 & -k_1 \\ \dfrac{k_0^2-s}{k_1} & k_0 \end{pmatrix}.$$

Имеет место следующее

Утверждение 5. *Из массива* $A_1(2, \lambda, \lambda)$, *построенного по формуле* $y=xM_k+\ell$, *где* M_k *— матрица Холла с характеристическим много-*

членом $f(x) = x^2 - rx - s$, *неприводимым над полем* $\mathrm{GF}(p^n)$, *при* $r = 0$ *можно выделить массив* $PA_1(2, \lambda, \lambda)$ $(\lambda = p^n = q)$.

Согласно доказанной теореме случай при $r = 0$ является простым, так как инволюция единственна и для $U(2)$-стойкого шифра необходимо брать только матрицы вида M_k, то есть «выбрасывая» для каждой из них противоположную вида $(-M_k)$. Теперь рассмотрим случай, когда $r \neq 0$.

Известно, что $U(2)$-стойкий шифр в системе Холла с $r \neq 0$ выделяется тогда и только тогда, когда в разбиении наклонов имеем $k \in K$ тогда и только тогда, когда $\overline{k} \notin K$, где

$$\det(M_k + M_{\overline{k}}) = \begin{vmatrix} k_0 + \overline{k}_0 & k_1 + \overline{k}_1 \\ \dfrac{-k_0^2 + rk_0 + s}{k_1} + \dfrac{-\overline{k}_0^2 + r\overline{k}_0 + s}{\overline{k}_1} & -(k_0 + \overline{k}_0) + 2r \end{vmatrix} =$$

$$= \varepsilon^2(-k_0 + rk_0 + s) + (2k_0\overline{k}_0 - r(k_0 + \overline{k}_0) + 2s)\varepsilon + (-\overline{k}_0 + r\overline{k}_0 + s) = 0$$

(здесь обозначено $\varepsilon = \overline{k}_1/k_1 \neq 0$, $\lambda = q$ нечетно), что равносильно равенству

$$\left(\varepsilon k_0 - \overline{k}_0\right)^2 - r(\varepsilon - 1)\left(\varepsilon k_0 - \overline{k}_0\right) - (\varepsilon + 1)^2 s = 0.$$

Находя дискриминант D получившегося квадратного уравнения для $x = (\varepsilon k_0 - \overline{k}_0)$ в поле нечетной характеристики, $D = r^2(\varepsilon - 1)^2 + 4(\varepsilon + 1)^2 s$, заключаем, что для существования координат k_0, $\overline{k}_0$ ключа необходимо, чтобы D был квадратом некоторого элемента t поля $\mathrm{GF}(q)$, то есть $D = t^2$. Так, при $\varepsilon = 1$ получаем $(k_0 - \overline{k}_0)^2 = 4s$.

Зададимся вопросом: существует ли разбиение K ключей на две части только по координате $k_1 \neq 0$, то есть такое, что если $k = (k_0, k_1) \in K$, то $(k_0', k_1') \in K$ при $k_1' = k_1''$? Для этого необходимо и достаточно, чтобы вместе с каждым $k_1 \neq 0$ существовало не менее $(q - 1)/2$ значений $k_1' = \varepsilon k \in K$ с такими ε, что D — не квадрат. Отсюда вытекает, что этих значений должно быть ровно $(q - 1)/2$. Более того, рассматривая элементы $k \in K$, $\varepsilon k \in K$, $\varepsilon^2 k \in K$, ..., видим, что ε порождает все квадраты в $\mathrm{GF}(q)^*$. А из того, что $D = 4r^2$ при $\varepsilon = -1$, ясно, что -1 должен быть неквадратом в $\mathrm{GF}(q)$, откуда $q \equiv 3 \pmod 4$. Итак, при $\varepsilon = u^2$ дискриминант

$$D = r^2(u^2 - 1)^2 + 4(u^2 + 1)^2 s$$

должен быть неквадратом для любого $u \in \mathrm{GF}(q)$.

Сделав дробно-линейное преобразование

$$t = \frac{u^2 - 1}{u^2 + 1} \longleftrightarrow \frac{1 + t}{1 - t},$$

находим элементы $t_1, t_2, \ldots, t_{(q-1)/2} \in \mathrm{GF}(q) \setminus \{1, -1\}$ такие, что $\dfrac{1+t_j}{1-t_j} = u_j^2$ суть все различные ненулевые квадраты поля $\mathrm{GF}(q)$ при $j = 1, 2, \ldots, (q-1)/2$. Добавив $t_0 = -1$, получим нулевой квадрат $u_0^2 = 0$, итого $(q-1)/2$ всех квадратов. Запишем D в виде

$$D = D_j = 4(u_j^2 + 1)^2\left(s + \frac{r^2}{4}t_j^2\right)$$

и приходим при $r \neq 0$ к необходимости того, что $s + v_j^2$ есть неквадрат для $(q-1)/2$ значений $v_j = rt_j/2$. При $t_j = 0 = v_j$ получаем, что s — неквадрат, то есть $s = -a^2$, так как -1 — не квадрат при $q \equiv 3 \pmod 4$. Значит,

$$s + v_j^2 = a^2\left(\left(\frac{v_j}{a}\right)^2 - 1\right)$$

и существуют такие элементы $\eta_j \neq 0$ $(j = 0, 1, \ldots, (q-1)/2)$, что

$$\left(\left(\frac{v_j}{a}\right)^2 - 1\right) = -\eta_j^2,$$

то есть $\xi_j^2 + \eta_j^2 = 1$, где $\xi_j = v_j/a$.

Осталось только подсчитать количество точек на кривой второго порядка («окружности») $\xi_j^2 + \eta_j^2 = 1$ над полем $\mathrm{GF}(q)$ при $q \equiv 3 \pmod 4$. Взяв ее рациональную параметризацию $\xi = 2\tau/\left(\tau^2 + 1\right)$, $\eta = \left(\tau^2 - 1\right)/\left(\tau^2 + 1\right)$ $(\tau \in \mathrm{GF}(q) \cup \{\infty\})$ видим, что на этой кривой всего $(q+1)$ точек, а с ненулевой ординатой η всего $(q-1)$ точек, следовательно ненулевых таких квадратов всего $(q-1)/2$ штук, вместо требуемых $(q+1)/2$. Итак, доказано, что такого разбиения множества ключей не существует.

Отметим, что если q — простое число, то разбиение ключей для $U(2)$-стойкого шифра при $r \neq 0$ возможно только вида, указанного выше, так как вычисление инволюции обратного наклона дает отображение $(k_0, k_1) \mapsto (k_0 + 2r, k_1)$, и вместе с ключом (k_0, k_1) в K содержится любой ключ (k_0', k_1) в силу простоты поля, потому что при $k_0' = (k_0 + 2rn) \bmod q$ ключ (k_0', k_1) получается из ключа (k_0, k_1) после n отображений. Итак, доказано

Утверждение 6. *При простом $q = p$ в системе Холла над полем* $\mathrm{GF}(q)$ *нет $U(2)$-стойкого подшифра, если $r \neq 0$.*

Также справедливо следующее

Утверждение 7. *Матрицы Холла, построенные с помощью неприводимого над* $\mathrm{GF}(3)$ *многочлена $f(x) = x^2 + 1$, образуют группу относительно операции умножения, эта система Холла изоморфна почти-полю $K(9)$.*

При доказательстве второй части утверждения 7 был найден один из изоморфизмов $\varphi(x)\colon\ \mathrm{Hall}(3) \to K(9)$, который задается следующей подстановкой:

$$\varphi(x) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 1 & 0 & 2 & 3 & 4 & 7 & 6 \end{pmatrix}.$$

Верхняя строка подстановки — это двумерные вектора системы Холла в десятичном представлении (например, $4_{10} = 11_3$). Нижняя строка — это степени первообразного элемента в почти-поле ($3 \equiv z^3$). Согласно наличию изоморфизма $\varphi(x)\colon M_9 \to K(9)$ [5, теорема 2], где M_9 — подгруппа группы Матьё M_{11}, также существует связь между системой Холла и группами Матьё. Данный результат позволит установить связь совершенных шифров с другими типами конечных плоскостей.

По $O(2)$-стойким шифрам был также получен следующий результат.

Утверждение 8. *$O(2)$-стойкий шифр, построенный в почти-поле $K(9)$, является линейным, но не билинейным и поэтому немультипликативным.*

Доказательство. Рассмотрим операцию умножения в почти-поле $K(9)$ как линейную операцию:

$$x \circ k = xM_k$$

и найдем такие матрицы M_k.

Пусть

$$M_k = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

где $a, b, c, d \in \mathrm{GF}(3)$.

Возьмем $x_1 = z^0 = (0, 1)$, $x_2 = z^1 = (1, 0)$, $y_1 = x_1 M_k = (e, f)$, $y_2 = x_2 M_k = (g, h)$ и составим систему уравнений:

$$\begin{cases} y_1 = x_1 M_k, \\ y_2 = x_2 M_k \end{cases} \Longrightarrow \begin{cases} (0, 1)\begin{pmatrix} a & b \\ c & d \end{pmatrix} = (e, f), \\ (1, 0)\begin{pmatrix} a & b \\ c & d \end{pmatrix} = (g, h) \end{cases} \Longrightarrow$$

$$\Longrightarrow \begin{cases} c = e, \\ d = f, \\ a = g, \\ b = h \end{cases} \Longrightarrow M_k = \begin{pmatrix} g & h \\ e & f \end{pmatrix}.$$

Так как

$$z^0 \circ z^t = z^t,\ z^t = k = (k_0, k_1),$$

то

$$M_k = \begin{pmatrix} g & h \\ k_0 & k_1 \end{pmatrix}.$$

Таким образом, получим следующие матрицы:

$$M_0 = E = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad M_1 = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}, \quad M_3 = \begin{pmatrix} 2 & 2 \\ 2 & 1 \end{pmatrix},$$
$$M_4 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \quad M_5 = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}, \quad M_6 = \begin{pmatrix} 1 & 2 \\ 2 & 2 \end{pmatrix}, \quad M_7 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}.$$

Проверим линейность таких матриц по ключу: для двух матриц, соответствующих двум базисным векторам $k = e = (0, 1)$ и $k = (1, 0)$ должно выполняться соотношение

$$\forall\, \alpha, \beta \in \mathrm{GF}(3) : \alpha M_0 + \beta M_1 = M_{\alpha+\beta}.$$

Проверим, взяв $\alpha = 2$, $\beta = 1$:

$$2M_0 + M_1 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} + \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 1 & 2 \end{pmatrix}.$$

Получившаяся матрица не принадлежит построенному выше множеству матриц M_k. □

Перейдем к проблеме построения $O(3)$-стойких шифров. Классическим примером такого шифра является его представление в группе $\mathrm{PGL}(2, \lambda)$ в силу ее точно 3-транзитивности [1, 9] в виде дробно-линейных преобразований [7, 10]. В [5] найден класс $O(3)$-стойких шифров на основе дробно-линейных подстановок в почти-полях. Дробно-линейная функция вида $y = (x + d) \setminus (xa + b)$, имеющая ясный геометрический смысл, может быть использована для построения $O(3)$-стойкого шифра в силу следующего утверждения.

Утверждение 9. *Функция $y = (x + d) \setminus (xa + b)$, где деление происходит в квазигруппе ненулевых элементов, является подстановкой тогда и только тогда, когда $b \neq da$.*

Доказательство. Подставим точки (x_1, y_1) и (x_2, y_2) в функцию зашифрования и составим систему

$$\begin{cases} (x_1 + d)y_1 = x_1a + b, \\ (x_2 + d)y_2 = x_2a + b \end{cases} \Longrightarrow (x_1 + d)y_1 - (x_2 + d)y_2 = x_1a + b - x_2a + b \Longrightarrow$$
$$\Longrightarrow (x_1 + d)y_1 - (x_2 + d)y_2 = (x_1 - x_2)a.$$

Если это не перестановка, то при $x_1 \neq x_2$ & $y_1 = y_2 = y$ получим $(x_1 - x_2)y = (x_1 - x_2)a$. Так как $x_1 - x_2 \neq 0$, то

$$y = a \implies (x + d)a = xa + b \iff xa + da = xa + b \iff da = b.$$

Значит, функция является перестановкой при условии, что $b \neq da$.

При $b = da$ получим

$$\begin{cases} (x_1 + d)y_1 = x_1a + da, \\ (x_2 + d)y_2 = x_2a + da \end{cases} \implies \begin{cases} (x_1 + d)y_1 = (x_1 + d)a, \\ (x_2 + d)y_2 = (x_2 + d)a \end{cases} \implies$$

$$\implies y_1 = y_2 = a;$$

это означает, что при $b = da$ для любого x будет справедливо соотношение $(x + d) \setminus (xa + b) = a$. □

Условие $b \neq da$ — аналог условия неравенства нулю определителя дробно-линейной функции.

Таким образом, в работе использованы методы конечной геометрии, с помощью которых была доказана теорема о взаимосвязи эндоморфных $U(2)$- и $O(2)$-стойких шифров с минимальным числом ключей в системе Веблена—Веддерберна; получены некоторые следствия из нее и рассмотрен пример применения данной теоремы на системе Холла. Установлена связь системы Холла над полем GF(3) с почти-полем $K(9)$. Доказана возможность использования дробно-линейной функции определенного вида для построения $O(3)$-стойкого шифра.

Авторы благодарят М. М. Глухова и М. А. Пудовкину за внимание к работе.

Литература

[1] *Зубов А. Ю.* Криптографические методы защиты информации. Совершенные шифры: Учебное пособие. М.: Гелиос АРВ, 2005. 192 с.

[2] *Коновалова С. С.*, *Титов С. С.* О конструкциях эндоморфных совершенных шифров // Материалы международной научной конференции по проблемам безопасности и противодействия терроризму. (Интеллектуальный Центр МГУ, 2–3 ноября 2005 г.) М.: МЦНМО, 2006. С. 168–180.

[3] *Гутарин Д. С.*, *Коновалова С. С.*, *Тимин В. И.*, *Титов Е. С.*, *Титов С. С.* Комбинаторные проблемы существования совершенных шифров // Труды Института математики и механики. Екатеринбург: УрО РАН. 2007. Т. 13. № 4. С. 61–73.

[4] *Konovalova S. S.*, *Titov S. S.* On construction of endomorphic perfect ciphers // Proceedings of Int. Security and Counteracting Terrorism Conference. Moscow, 2006. P. 179–191.

[5] *Коновалова С. С.*, *Титов С. С.* Построение $O(L)$- и $U(L)$-стойких шифров в конечных плоскостях // Материалы Третьей международной научной конференции по проблемам безопасности и противодействия терроризму. (МГУ им. М. В. Ломоносова, 25–27 октября 2007 г.) М.: МЦНМО, 2008. С. 191–209.

[6] *Коновалова С. С.*, *Титов С. С.* О конструкциях эндоморфных совершенных шифров // Сборник «Проблемы прикладной математики». Екатеринбург: Изд-во УрГУПС, 2005–2006. Т. 2, № 41(124). С. 70–106.

[7] *Артин Э.* Геометрическая алгебра. М.: Наука, 1969. 284 с.

[8] *Холл М.* Комбинаторика. М.: Мир, 1970. 424 с.

[9] *Глухов М. М.*, *Елизаров В. П.*, *Нечаев А. А.* Алгебра: учебник. М.: Гелиос АРВ. В 2-х т. Т. 2. 2003. 416 с.

[10] *Сидельников В. М.* Криптография и теория кодирования // Московский университет и развитие криптографии в России. (Материалы конференции в МГУ 17–18 октября 2002 г.) М.: МЦНМО, 2003. С. 49–84.

Дискретные временные ряды и их использование в задачах защиты информации

Ю. С. Харин

1. Введение

Удобной математической моделью выходных последовательностей криптосистем, ключевых последовательностей, входных последовательностей в системах криптоанализа, а также контейнеров и стего в стеганографических системах является дискретный временной ряд $x_t \in A$, т. е. случайный процесс с дискретным временем $t \in \mathbb{N} = \{1, 2, \ldots\}$ и конечным пространством состояний $A = \{0, 1, \ldots, N-1\}$ мощности $2 \leqslant N < +\infty$.

Для исследования и использования модели «непрерывного» временного ряда, когда $A = \mathbb{R}^N$, разработана ставшая классической теория статистического анализа временных рядов [1], базирующаяся на процессах второго порядка, гауссовских процессах, моделях ARIMA(p, d, q) и их обобщениях. В задачах защиты информации A — конечное множество, поэтому эти классические модели не применимы и актуальна проблема разработки специальных моделей дискретных временных рядов.

В статье приводится краткий аналитический обзор семейства моделей дискретных временных рядов, имеющих практическое значение для задач защиты информации, а также результатов по их идентификации.

2. Цепь Маркова s-го порядка

Определенный на вероятностном пространстве $(\Omega, \mathcal{F}, \mathsf{P})$ дискретный временной ряд $x_t \in A$, обладающий марковским свойством s-го порядка ($s \in \mathbb{N}$):

$$\begin{aligned} &\mathsf{P}\{x_t = i_t \,|\, x_{t-1} = i_{t-1}, \ldots, x_1 = i_1\} = \\ &\quad = \mathsf{P}\{x_t = i_t \,|\, x_{t-1} = i_{t-1}, \ldots, x_{t-s} = i_{t-s}\} = \\ &\quad = p_{i_{t-s}, \ldots, i_{t-1}, i_t}, \quad t > s,\ i_1, \ldots, i_t \in A, \end{aligned} \tag{1}$$

называется [2] (сложной) цепью Маркова s-го порядка (s). В условиях стационарности эта модель полностью определяется заданием $(s+1)$-мерной матрицы вероятностей одношаговых переходов $P = (p_{i_1,\dots,i_s,i_{s+1}})$ и является универсальной моделью, учитывающей зависимости «большой глубины». Однако для такой модели число параметров $D_{\text{ЦМ}(s)}$ растет экспоненциально при увеличении порядка s: $D_{\text{ЦМ}(s)} = N^s(N-1)$, и для ее идентификации требуется наблюдать реализацию $X_1^n = (x_1, \dots, x_n) \in A^n$ не всегда доступной на практике длительности $n > D_{\text{ЦМ}(s)}$.

Представим «малопараметрические» модели цепи Маркова высокого порядка s.

3. Модель Джекобса—Льюиса

Эта модель предложена в 1978 г. для экономических приложений и определяется стохастическим уравнением [3]:

$$x_t = \mu_t x_{t-\eta_t} + (1-\mu_t)\xi_t, \quad t > s, \tag{2}$$

где $\{\xi_t,\ \eta_t,\ \mu_t : t > s\}$, $\{x_1, \dots, x_s\}$ — независимые случайные величины,

$$\mathsf{P}\{\xi_t = i\} = \pi_i, \quad i \in A,\ t > s,\ \sum_{i\in A}\pi_i = 1;$$

$$\mathsf{P}\{\eta_t = j\} = \lambda_j, \quad j \in \{1, \dots, s\},\ t > s,\ \sum_{j=1}^{s}\lambda_j = 1,\ \lambda_s \neq 0;$$

$$\mathsf{P}\{\mu_t = 1\} = 1 - \mathsf{P}\{\mu_t = 0\} = \rho, \quad t > s;$$

$$\mathsf{P}\{x_t = i\} = \pi_i, \quad i \in A,\ t \leqslant s.$$

Число параметров этой модели линейно зависит от глубины памяти s: $d_{\text{JL}} = N + s - 1$. Соотношение (2) адекватно моделирует генератор Geffe со случайной обратной связью [4].

Отметим, что в [3] исследовались лишь простейшие вероятностные свойства модели (2), а задачи идентификации не решались. В [5] доказано, что (2) является (s) с матрицей переходов $P = (p_{i_1,\dots,i_{s+1}})$:

$$p_{i_1,\dots,i_{s+1}} = (1-\rho)\pi_{i_{s+1}} + \rho\sum_{j=1}^{s}\lambda_j\delta_{i_{s-j+1},i_{s+1}}, \quad i_1, \dots, i_{s+1} \in A,$$

где $\delta_{j,k}$ — символ Кронекера. В [5] построены состоятельные оценки параметров $\widetilde{\pi}$, $\widetilde{\rho}$, $\widetilde{\lambda}$ по X_1^n и с их помощью оценки максимального правдоподобия (ОМП) $\widehat{\pi}$, $\widehat{\rho}$, $\widehat{\lambda}$, а также критерий обобщенного отношения правдоподобия (асимптотического размера $\varepsilon \in (0,\ 1)$) для проверки гипотез

H_0: $\{\pi = \pi^0, \lambda = \lambda^0, \rho = \rho^0\}$, $H_1 = \overline{H}_0$, где π^0, λ^0, ρ^0 — некоторые заданные гипотетические значения параметров модели (2).

4. Дискретная авторегрессия DAR(s)

Модель DAR(s) является обобщением [6] модели Джекобса—Льюиса и определяется стохастическим разностным уравнением над полем $A = \mathrm{GF}(2) = \{0, 1\}$:

$$x_t = \alpha_s x_{t-1} \oplus \ldots \oplus \alpha_1 x_{t-s} \oplus \xi_t, \quad t > s, \tag{3}$$

где $\alpha = (\alpha_1, \ldots, \alpha_s) \in A^s$ — вектор коэффициентов авторегрессии ($\alpha_1 = 1$), $\{\xi_t\}$ — н. о. р. с. в. Бернулли, $\mathsf{P}\{\xi_t = 1\} = 1 - \mathsf{P}\{\xi_t = 0\} = p \in [0, 1/2)$, $X_1^s = (x_1, \ldots, x_s) \in A^s$ — вектор начальных значений. Число параметров модели $d_{\mathrm{DAR}(s)} = 2s + 1$.

ОМП параметров модели (3) определяются соотношениями:

$$\sum_{t=s+1}^{n} (x_t \oplus \alpha_s x_{t-1} \oplus \ldots \oplus \alpha_1 x_{t-s}) \to \min_{\alpha, X_1^s},$$

$$\widehat{p} = \frac{1}{n-s} \sum_{t=s+1}^{n} (x_t \oplus \alpha_s x_{t-1} \oplus \ldots \oplus \alpha_1 x_{t-s}).$$

5. MTD-модель Рафтери и ее обобщение

MTD (Mixture Transition Distribution)-модель предложена в 1985 г. А. Рафтери [7] и определяется следующим «малопараметрическим» видом матрицы (1):

$$p_{i_1,\ldots,i_s,i_{s+1}} = \sum_{j=1}^{s} \lambda_j q_{i_j,i_{s+1}}, \quad i_1, \ldots, i_{s+1} \in A, \tag{4}$$

где $Q = (q_{i,k})$ — некоторая стохастическая $(N \times N)$-матрица, $0 \leqslant q_{i,k} \leqslant 1$, $\sum\limits_{k \in A} q_{i,k} \equiv 1$, i, $k \in A$, $\lambda = (\lambda_1, \ldots, \lambda_s)'$ — некоторое дискретное распределение вероятностей ($\lambda_1 > 0$).

MTDg (обобщенная MTD)-модель задается обобщением (4):

$$p_{i_1,\ldots,i_s,i_{s+1}} = \sum_{j=1}^{s} \lambda_j q^{(j)}_{i_j,i_{s+1}}, \quad i_1, \ldots, i_{s+1} \in A, \tag{5}$$

где $Q^{(j)} = \left(q^{(j)}_{i,k}\right)$ — некоторая стохастическая матрица для j-го лага. Число параметров для MTDg $d_{\mathrm{MTDg}} = s(N(N-1)+1) - 1$.

В [5] установлены условия эргодичности MTD-модели (4), для MTDg-модели (5) построены состоятельные оценки параметров λ, $\{Q^{(j)}\}$, превосходящие по точности оценки из [7], а также тест проверки гипотез о значениях параметров модели (4).

6. Цепь Маркова s-го порядка с r частичными связями (s, r)

Эта «малопараметрическая» модель предложена в Белгосуниверситете в 2003 г. [8, 9]. Примем обозначения: $r \in \{1, \ldots, s\}$ — параметр, называемый числом связей; $M_r^0 = (m_1^0, \ldots, m_r^0) \in \mathcal{M}$ — произвольный целочисленный r-вектор с упорядоченными компонентами $1 = m_1^0 < m_2^0 < \ldots < m_r^0 \leqslant s$, называемый шаблоном связей; $\mathcal{M}$ — множество всевозможных таких векторов с r компонентами мощности $K = |\mathcal{M}| = \mathrm{C}_{s-1}^{r-1}$; $Q^0 = (q^0_{j_1,\ldots,j_{r+1}})$, $j_1, \ldots, j_{r+1} \in A$, — некоторая $(r+1)$-мерная стохастическая матрица.

Цепь Маркова x_t называется [8] цепью Маркова s-го порядка с r частичными связями и обозначается (s, r), если вероятности одношаговых переходов (1) имеют «малопараметрический» вид:

$$p_{i_1,\ldots,i_{s+1}} = q^0_{i_{m_1^0},\ldots,i_{m_r^0},i_{s+1}}, \quad i_1, \ldots, i_{s+1} \in A. \tag{6}$$

Соотношение (6) означает, что вероятность перехода процесса в состояние i_{s+1} в момент времени $t > s$ зависит не от всех s предыдущих состояний $i_1, \ldots, i_s$, а лишь от r избранных состояний $i_{m_1^0}, \ldots, i_{m_r^0}$. Число параметров $d_{\mathrm{ЦМ}(s,r)} = N^r(N-1) + r - 1$. Выигрыш в числе параметров может оказаться весьма существенным: например, если $N = 2$, $s = 32$, $r = 3$, то $D_{\mathrm{ЦМ}(s)} \approx \approx 4.1 \cdot 10^9$, в то время как $d_{\mathrm{ЦМ}(s,r)} = 10$.

Заметим, что если $r = s$, $M_r^0 = (1, \ldots, s)$, то $P = Q^0$ и $(s, s) = (s)$ есть цепь Маркова с полными связями. Конструктивным примером (s, r) является двоичная авторегрессия $\mathrm{DAR}(s)$ с r ненулевыми коэффициентами (3), частным случаем которой является линейная рекуррента над кольцом Z_2, порожденная многочленом степени s с r ненулевыми коэффициентами.

В [8, 9] для (s, r) решены следующие задачи: установлены достаточные условия эргодичности; построены состоятельные оценки параметров $\widehat{r} \in \{r_-, r_- + 1, \ldots, r_+\}$ $(r_+ < s)$, $\widehat{M}_{\widehat{r}}$, $\widehat{Q}$ по X_1^n; построен критерий проверки гипотез $H_0 : Q^0 = Q_0$ против альтернативы общего вида $H_1 = \overline{H}_0$, где Q_0 — некоторая заданная $(r+1)$-мерная стохастическая матрица; методом асимптотических разложений при $n \to +\infty$ получены оценки матрицы вариаций $\widehat{Q}$ и вероятности ошибки $\mathsf{P}\{\widehat{M}_r \neq M_r^0\}$.

7. Цепь Маркова переменного порядка ЦМПП

Модель ЦМПП предложена в 1999 г. П. Бюльманом [10]. Цепь Маркова x_t называется [10] цепью Маркова переменного порядка $l = l(j_1, \ldots, j_s)$: $A^s \to \{1, \ldots, s\}$, если вероятности одношаговых переходов (1) имеют специальный «малопараметрический» вид:

$$p_{j_1,\ldots,j_{s+1}} = q_{j_{s-l+1},\ldots,j_{s+1}}, \quad j_1, \ldots, j_{s+1} \in A. \tag{7}$$

При этом дискретная функция $c(J_1^s) = J_{s-l+1}^s : A^s \to A^l$, присутствующая в индексном выражении (7), называется контекстной функцией, а дерево $\tau = \{u : u = c(J_1^s),\ J_1^s \in A^s\}$ — контекстным деревом. Число параметров для модели ЦМПП $d_{\text{ЦМПП}} = |\tau|(N-1)$.

В [10] предложен алгоритм оценивания контекстного дерева по наблюдениям $X_1^n = (x_1, \ldots, x_n) \in A^n$.

8. INAR(s)-модель

Эта модель предложена в 2006 г. [11] на основе модификации модели Эль-Заида и Эль-Оша для целочисленных авторегрессионных временных рядов $z_t \in Z_+ = \{0, 1, \ldots\}$ в экономических приложениях:

$$x_t = \left(\sum_{i=1}^{s} \theta_i \sum_{j=0}^{x_{t-i}} \xi_{(t-1)N+j}^{(i)} \right) \bmod N, \quad t > s, \tag{8}$$

где $\{\xi_k^{(i)} \in \{0, 1\} : k \in \mathbb{N},\ i = 1, 2, \ldots, s\}$ — н. о. р. с. в. Бернулли, $\theta = (\theta_1, \ldots, \theta_s) \in A^s$ — вектор коэффициентов. Число параметров: $d_{\text{INAR}(s)} = s + 1$. Модель INAR($s$) удобна для моделирования выходных последовательностей «комбинирующих» генераторов [4].

В [11] установлены условия эргодичности модели (8) и найдены уклонения вероятностей одношаговых переходов и стационарного распределения от равномерного распределения вероятностей.

Результаты численных экспериментов [12, 13] показывают достаточную эффективность идентификации приведенных выше «малопараметрических» моделей дискретных временных рядов.

Литература

[1] *Андерсон Т. В.* Статистический анализ временных рядов. М.: Мир, 1976.

[2] *Дуб Дж.* Вероятностные процессы. М.: ФМ, 1956.

[3] *Jacobs P. A.*, *Lewis P. A. W.* Discrete time series generated by mixtures // J. Royal Statist. Soc. Ser. B. 1978. V. 40, № 1. P. 94–105.

[4] *Алферов А. П.*, *Зубов А. Ю.*, *Кузьмин А. С.*, *Черемушкин А. В.* Основы криптографии. М.: Гелиос АРВ, 2001.

[5] *Харин Ю. С.* Вероятностно-статистический анализ цепей Маркова высокого порядка // Вестник БГУ. Сер. 1. 2006, № 3. С. 80–86.

[6] *Максимов Ю. И.* О цепях Маркова, связанных с двоичными регистрами сдвига со случайными элементами // Труды по дискретной математике. 1997. Т. 1. С. 203–220.

[7] *Raftery A. E.* A model for high-order Markov chains // J. Royal Statist. Soc. Ser. B. 1985. V. 47, № 1. P. 528–539.

[8] *Харин Ю. С.* Цепи Маркова с r-частичными связями и их статистическое оценивание // Доклады НАН Беларуси. 2004. Т. 48, № 1. С. 40–44.

[9] *Харин Ю. С.*, *Петлицкий А. И.* Цепь Маркова s-го порядка с r частичными связями и статистические выводы о ее параметрах // Дискретная математика. 2007. Т. 19, № 2. С. 109–130.

[10] *Buhlmann P.*,*Wyner A. J.* Variable length Markov chains // The Annals of Statistics. 1999. V. 27, № 2. P. 480–513.

[11] *Ярмола*, *А. Н.*, *Харин Ю. С.* INAR-последовательности и их применение в задачах защиты информации // Материалы Второй международной научной конференции по проблемам безопасности и противодействия терроризму (МГУ, 25–26 октября 2006 г.). М.: МЦНМО, 2007, с. 276–280.

[12] *Харин Ю. С.* Оптимальность и робастность в статистическом прогнозировании. Мн.: БГУ, 2008.

[13] *Харин Ю. С.*, *Агиевич С. В.*, *Микулич Н. Д.* О стандартизации алгоритмов криптографической защиты информации // Управление защитой информации. 2008. Т. 12, № 1. С. 84–88.

О некоторых свойствах линейных кодов, образующих носители корреляционно-иммунных булевых функций

Е. К. Алексеев

Аннотация

В работе рассматриваются некоторые свойства линейных кодов, образующих носители корреляционно-иммунных булевых функций. Вводится понятие минимальной корреляционно-иммунной функции. Доказывается теорема о количестве корреляционно-иммунных функций веса 4.

Ключевые слова: корреляционная иммунность, булева функция, линейный код, носитель булевой функции.

1. Введение

Корреляционно-иммунные функции являются важным строительным блоком при синтезе поточных шифров. Эти функции являются хорошо известным объектом в таких разделах математики, как комбинаторный анализ и теория кодирования.

В данной работе рассматриваются некоторые связи корреляционно-иммунных функций с теорией кодирования. Рассматриваются линейные коды, образующие носители таких функций. Вводится понятие минимальной корреляционно-иммунной функции. Приводится формула для мощности множества корреляционно-иммунных функций веса 4.

2. Основные определения и обозначения

Пусть $F_2 = \mathrm{GF}(2)$, $V_n = F_2^n$. Пусть $\mathcal{F}_n = \{f \mid f\colon V_n \to F_2\}$.

Определение 1. *Линейный код C длины n* — это линейное подпространство векторного пространства V_n.

Определение 2. *Размерностью k_C* кода C называется его размерность как векторного пространства $k_C = \dim C$.

Определение 3. *Минимальным расстоянием* $d(C)$ кода C называется минимальное расстояние Хемминга между кодовыми словами

$$d(C) = \min\{\operatorname{dist}(c,\, c') \mid c,\, c' \in C,\ c \neq c'\}.$$

Определение 4. *Носителем* булевой функции $f \in \mathcal{F}_n$ называется множество $1_f = \{x \in V_n \mid f(x) = 1\}$.

Определение 5. *Индикаторной функцией* множества $S \subseteq V_n$ называется такая функция $f = I_S \in \mathcal{F}_n$, что $f(x) = 1$ тогда и только тогда, когда $x \in S$.

Определение 6. Булева функция $f(x^{(1)},\, x^{(2)},\, \ldots,\, x^{(n)}) \in \mathcal{F}_n$ называется *корреляционно-иммунной* порядка m, $0 < m \leqslant n$, если для любых наборов $1 \leqslant i_1 < \ldots < i_m \leqslant n$, $a^{(i)} \in F_2$, $j = 1,\, \ldots,\, m$, выполняются соотношения

$$\operatorname{wt}\bigl(f^{a^{(1)},\ldots,a^{(m)}}_{i_1,\ldots,i_m}\bigr) = \frac{\operatorname{wt}(f)}{2^m}.$$

Определение 7. *Порядком корреляционной иммунности* называется число

$$\operatorname{cor} f = \max\{m \in \mathbb{N} \mid f \text{корреляционно-иммунна порядка } m\}.$$

Определение 8. *Преобразованием Уолша—Адамара* булевой функции $f \in \mathcal{F}_n$ называется целочисленная функция на V_n, определяемая следующим равенством:

$$W_f(u) = \sum_{x \in V_n} (-1)^{f(x) \oplus \langle x,u \rangle}.$$

Существует критерий того, что функция корреляционно-иммунна порядка m (См. [1].).

Теорема 1. *Булева функция $f \in \mathcal{F}_n$ корреляционно-иммунна порядка m тогда и только тогда, когда $W_f(u) = 0$ для всех векторов $u \in V_n$ таких, что $1 \leqslant \operatorname{wt}(u) \leqslant m$.*

Определение 9. $\operatorname{CI}(n) = \{f \in \mathcal{F}_n \mid \operatorname{cor} f \geqslant 1\}$.

Определение 10. $\operatorname{Mir}(n) = \{f \in \mathcal{F}_n \mid \forall\, x \in V_n\ f(x) = f(x \oplus \overline{1})\}$.

Определение 11. *Ортогональной таблицей* $\operatorname{OA}_\nu(m,\, n,\, 2,\, t)$ размера $m \times n$ с ограничениями, уровня 2, силы t и индекса ν называется $(m \times n)$-матрица M над полем F_2, обладающая свойством: в любом подмножестве из t столбцов матрицы M любой из 2^t векторов пространства V_t встречается как строка ровно ν раз.

Теорема 2. *Для функции $f \in \mathcal{F}_n$ выполнено неравенство $\operatorname{cor} f \geqslant t$ тогда и только тогда, когда ее таблица истинности M_f ($\operatorname{wt}(f) \times n$ — матрица, строками которой являются векторы из V_n, значение*

функции на которых равно 1) *является ортогональной таблицей* $\mathrm{OA}_{\nu}(\mathrm{wt}(f), n, 2, t)$.

Неравенство Рао для $\mathrm{OA}(N, k, s, t = 2u)$:

$$N \geqslant \sum_{i=0}^{u} \binom{k}{i} (s-1)^i.$$

Для случая $u = 1$ и $s = 2$, получаем неравенство $N \geqslant 1 + n$.

Определение 12. *Двоичный код Хемминга* H_r длины $n = 2^r - 1$ $(r > 1)$ имеет проверочную матрицу H, столбцы которой состоят из всех ненулевых двоичных векторов длины r, причем каждый вектор встречается в матрице один раз. Код H_r имеет параметры $n = 2^r - 1$, $k = 2^r - 1 - r$, $d = 3$.

Определение 13. Два двоичных кода называются *эквивалентными*, если они отличаются только перестановкой координат.

Определение 14. Если $S(n)$ обозначает некоторое подмножество булевых функций от n переменных, то через $S(n, k)$, где $0 \leqslant k \leqslant n$, будем обозначать множество $\{f \in S(n) \mid \mathrm{wt}(f) = k\}$.

3. Минимальность корреляционно-иммунных функций

Определение 15. Функция $f \in \mathrm{CI}(n)$ называется *минимальной корреляционно-иммунной функцией*, если не существует функции $g \in \mathrm{CI}(n)$ такой, что $1_g \subset 1_f$.

Известно, что если $f, g \in \mathrm{CI}(n)$ и $f \cdot g \equiv 0$, то $f \oplus g \in \mathrm{CI}(n)$.

Предложение 1. *Функция* $f \in \mathrm{CI}(n)$ *является минимальной корреляционно-иммунной функцией тогда и только тогда, когда* f *нельзя разложить в сумму двух ортогональных корреляционно-иммунных функций* (*т.е. в сумму двух функций* $g, h \in \mathrm{CI}(n)$ *таких, что* $g \cdot h \equiv 0$).

Доказательство. Вытекает непосредственно из определения 15. □

В работе [2] доказывается, что $\mathrm{CI}(n) = \bigcup_{g \in \mathrm{BCI}(n)} (g \oplus \mathrm{Mir}(n)|_g)$, где

$$\mathrm{BCI}(n) = \{f \in \mathrm{CI}(n) \mid f(x) \cdot f(x \oplus 1) = 0 \ \forall\, x \in V_n\},$$

$$\mathrm{Mir}(n)|_g = \left\{ f \in \mathrm{Mir}(n) \;\middle|\; 1_f \subseteq 1_{g(x) \oplus g\left(x \oplus \widetilde{1}\right) \oplus 1} \right\}.$$

Пример 1. Минимальными корреляционно-иммунными булевыми функциями веса 2 являются функции из множества $\mathrm{Mir}(n, 2)$.

Пример 2. Так как не существует $f \in \mathrm{Mir}(n, 2)$ такой, что $1_f \subset 1_g$, где $g \in \mathrm{BCI}(n, 4)$, то все функции из $\mathrm{BCI}(n, 4)$ являются минимальными веса 4.

Пример 3. Так как для любой функции $g \in \mathrm{BCI}(n)$ такой, что $\mathrm{wt}(g) = 6$, не существует функции $g \in \mathrm{CI}(n)$ такой, что $1_g \subset 1_f$, то любая функция из $\mathrm{BCI}(n, 6)$ является минимальной корреляционно-иммунной функцией веса 6.

Понятно, учитывая разложение $\mathrm{CI}(n)$ по подпространствам $\mathrm{Mir}(n)|_g$, что множествами $\mathrm{Mir}(n, 2)$, $\mathrm{BCI}(n, 4)$ и $\mathrm{BCI}(n, 6)$ исчерпываются все минимальные корреляционно-иммунные функции веса 2, 4 и 6. Пользуясь этим же разложением можно сказать, что любая минимальная корреляционно-иммунная функция f веса больше чем 2 принадлежит множеству $\mathrm{BCI}(n)$.

Введем обозначение:

$$\mathrm{MCI}(n) = \{f \in \mathrm{CI}(n) \mid f \text{ — минимальная корреляционно-иммунная булева функция}\}.$$

4. Построение корреляционно-иммунных функций с использованием линейных кодов

В книге [3] рассматривается вопрос построения ортогональных массивов с помощью кодов исправляющих ошибки. В терминах булевых функций результат, описанный в книге, можно сформулировать следующим образом.

Теорема 3. *Пусть* $L < V_n$. *Тогда* $\mathrm{cor}\, I_L = r \iff d(L^\perp) = r + 1$.

Доказательство. Заметим, что функция $S_u(x) = \langle u, x\rangle$ либо тождественно равна нулю, либо уравновешенна на L.

Рассмотрим спектр коэффициентов Фурье функции $f = I_L$.

$$F_{I_L}(u) = \sum_{x \in L} (-1)^{\langle u,x\rangle} = \sum_{x \in L} (-1)^{S_u(x)}.$$

1. Предположим, что $d(L^\perp) = r + 1$. Тогда $\exists\, u \in L^\perp : \mathrm{wt}(u) = r + 1$ и $\forall\, u \in V_n : \mathrm{wt}(u) \leqslant r$ $\langle u, x\rangle = S_u(x)$ — уравновешенна. Следовательно, $\forall\, u \in V_n : 1 \leqslant \mathrm{wt}(u) \leqslant r$ $F_{I_L}(u) = 0 \implies \mathrm{cor}\, I_L = r$.
2. Предположим, что $\mathrm{cor}\, I_L = r$. Следовательно, $\forall\, u \in V_n : 1 \leqslant \mathrm{wt}(u) \leqslant r$ $S_u(x)$ уравновешенна на L. И $\exists\, u \in V_n : \mathrm{wt}(u) = r + 1$, $S_u(x) = 0$ на $L \implies d(L^\perp) = r + 1$. □

Обозначим через $\mathrm{LCI}(n)$ множество корреляционно-иммунных функций от n переменных, которые являются индикаторными функциями множеств $L \oplus z$, где $L < V_n$, а $z \in V_n$.

Известно, что если $L < V_n$, то $deg I_L = n - \dim L$.

Определение 16. Функция $f \in \mathrm{LCI}(n)$ называется *минимальной* в классе $\mathrm{LCI}(n)$, если не существует такой функции $g \in \mathrm{LCI}(n)$, что $1_g \subset 1_f$.

Предложение 2. *$f = I_L \in \mathrm{LCI}(n)$ является минимальной в классе $\mathrm{LCI}(n) \iff L^\perp$ нельзя расширить с сохранением свойства $d_{L^\perp} > 1$.*

Доказательство. Необходимость. Очевидно, что $L^\perp < L'^\perp \Longrightarrow L > L'$. Если бы $L_\perp$ можно было бы расширить с сохранением $d_{L^\perp} \neq 1$, то существовало бы $L' < L : d_{L'^\perp} > 1 \Longrightarrow I_{L^\perp}$ была бы $< I_L$ и $I_{L'} \in \mathrm{LCI}(n)$. Получили противоречие, которое доказывает необходимость.

Достаточность. Очевидно, что $\exists L' < L \Longrightarrow \exists L^\perp < L'^\perp$. Если бы $I_{L'} \in \mathrm{CI}(n)$, то $d(L'^\perp) > 1 \Longrightarrow$ можно было бы расширить $L^\perp$ с сохранением свойства $d(L'^\perp) > 1$. Получили противоречие, которое доказывает достаточность. □

Предложение 3. *Если $L < V_n$ такова, что I_L является минимальной корреляционно-иммунной функцией в классе $\mathrm{LCI}(n)$, то выполнено неравенство*

$$\dim L \leqslant \log_2(n+1).$$

Доказательство. Если $\dim L = m$, то $\dim L^\perp = n - m$. V_n состоит из 2^m различных сдвигов пространства $L^\perp$ на векторы из V_n. Т.к. I_L является минимальной в $\mathrm{LCI}(n)$, то $L^\perp$ нельзя расширить так, чтобы для расширенной плоскости $L_1^\perp$ сохранялось свойство $d(L_1^\perp) > 1$. Следовательно, при любом сдвиге пространства $L^\perp$ возникает плоскость, в которой присутствуют векторы веса 1. Следовательно, все сдвиги присутствуют в множестве сдвигов на векторы $e_1, \ldots, e_n$ веса 1. Поэтому $2^m \leqslant n+1 \Longrightarrow m = \dim L \leqslant \log_2(n+1)$. □

Предложение 4. *Для любой функции $f = I_L$ минимальной в классе $\mathrm{LCI}(n)$, выполнено неравенство $\mathrm{cor}\, f \leqslant 2$.*

Доказательство. Пусть $\mathrm{cor}\, f \geqslant 3$, тогда $d(L^\perp) \geqslant 4$. Следовательно, $L^\perp$ можно расширить с помощью любого вектора веса 2 с сохранением свойства $d(L^\perp) > 1$. □

Возможен ли случай, когда для функции $f = I_L$ минимальной в $\mathrm{LCI}(n)$ выполнено равенство $\mathrm{cor}\, f = 2$? Рассмотрим функцию $f = I_L$, которая является минимальной в классе $\mathrm{LCI}(n)$. Для нее выполнено неравенство $\dim L \leqslant \log_2(n+1) \Longrightarrow \#L = \mathrm{wt}(I_L) = 2^{\dim L} \leqslant n+1$. Если эта функция является корреляционно-иммунной порядка 2, то для нее верно неравенство Рао: $\mathrm{wt}(f) \geqslant n+1$. Учитывая предыдущее неравенство, получаем, что минимальная в $\mathrm{LCI}(n)$ функция I_L может быть корреляционно-иммунной порядка 2 в том и только том случае, когда $\mathrm{wt}(I_L) = 2^{\dim L} = n+1$. Пусть

$\dim L = r$. Тогда пространство $L^{\perp}$ имеет параметры $n = 2^r - 1$, $\dim L^{\perp} = 2^r - 1 - r$, $d(L^{\perp}) = 3$. Но любой линейный код с такими параметрами эквивалентен коду Хемминга.

Таким образом, для всех функций $f = I_L \in \mathrm{LCI}(n)$ таких, что $\mathrm{cor}\, f = 2$, являющихся минимальными в этом классе, код $L^{\perp}$ эквивалентен коду Хемминга.

5. Корреляционно-иммунные функции веса 4

Предложение 5. *Любая функция $f \in \mathrm{BCI}(n, 4)$ представима в виде $f = I_{L \oplus z}$ для некоторого подпространства $L < V_n$ такого, что $\dim L = 2$, и вектора $z \in V_n$.*

Доказательство. Пусть $\overline{L} = \{\overline{x}_1, \overline{x}_2, \overline{x}_3, \overline{x}_4\}$, где $\overline{x}_i \in 1_{\overline{f}}$, $i = 1, 2, 3, 4$ для $\overline{f} \in \mathrm{BCI}(n, 4) \Longrightarrow \overline{f}(x \oplus \overline{x}_i) \in \mathrm{BCI}(n, 4)$ для любого $\overline{x}_i$, $i = 1, 2, 3, 4$.

Рассмотрим $L = \overline{L} \oplus \overline{x_1} = \{\overline{0}, x_1, x_2, x_3\}$. Тогда

$$f = I_L = \begin{cases} 1 & \text{при } x \in L, \\ 0 & \text{при } x \notin L \end{cases}$$

и $f \in \mathrm{BCI}(n, 4)$.

Так как $f \in \mathrm{BCI}(n)$, то $x_i \oplus x_j \neq \widetilde{1}$. Рассмотрим x_1, x_2, x_3 и $\widetilde{0} = x_4 : x_i = (x_i^1, x_i^2, \ldots, x_i^n)$. Очевидно, что если $x_1^j = 0$, то $x_2^j = x_3^j = 1$, а если $x_1^j = 1$, то $x_2^j \oplus x_3^j = 1$. Следовательно, $x_2 \oplus x_3 = x_1 \Longrightarrow L$ является линейным подпространством пространства V_n. Поэтому для любой $f \in \mathrm{BCI}(n, 4)$ $\exists\, L < V_n$ и $\exists\, z \in V_n$ такие, что $f = I_{L \oplus z}$. □

Пространство $L < V_n$, которое является носителем функции $f \in \mathrm{BCI}(n, 4)$, удовлетворяет следующим свойствам: $\#L = 4$ и $\forall\, x, y \in L$ $x \oplus y \neq \widetilde{1}$ и $x_1 \vee x_2 \vee x_3 \vee x_4 = \widetilde{1}$.

Предложение 6. $\#\,\mathrm{BCI}(n, 4) = 2^{n-3}(3^{n-1} - 2^n + 1)$.

Доказательство. Пусть $L < V_n$ и $L = \{x_0 = \overline{0}, x_1, x_2, x_3\}$. Не ограничивая общности, положим: $x_1 = (0, x_1^2, \ldots, x_1^n)$, $x_2 = (1, x_2^2, \ldots, x_2^n)$ и $x_3 = (1, x_3^2, \ldots, x_3^n)$.

Обозначим через $\widetilde{x}_i^T$ столбец x_i^T без первого значения x_i^1. В столбце $\widetilde{x}_1^T$ может быть k нулей при $0 \leqslant k \leqslant n - 2$. Фиксировав столбец $\widetilde{x}_1^T$, мы получим ограничения на столбцы $\widetilde{x}_2^T$ и $\widetilde{x}_3^T$. В тех строках, в которых в столбце $\widetilde{x}_1^T$ стоят нули, должны стоять значения 1 в столбцах $\widetilde{x}_2^T$ и $\widetilde{x}_3^T$. В тех строках, в которых в $\widetilde{x}_1^T$ стоят единицы, должны стоять противоположные значения в столбцах $\widetilde{x}_2^T$ и $\widetilde{x}_3^T$. Таким образом, после фиксации значения $\widetilde{x}_1^T$ однозначно определяются значения в k строках столбцов $\widetilde{x}_2^T$ и $\widetilde{x}_3^T$.

Понятно, что та часть столбца $\widetilde{x}_2^T$, которая осталась незафиксированной, однозначно определяет значение той части строк, которая осталась нефиксированной в столбце $\widetilde{x}_3^T$. Если выделить нефиксированную часть столбцов $\widetilde{x}_2^T$ и $\widetilde{x}_3^T$, то получим векторы X_2^T и X_3^T такие, что $X_2^T = X_3^T \oplus \overline{1}$.

Векторы X_2^T и X_3^T принадлежат V_{n-1-k}. X_2^T не может принимать значения $\overline{0}$ и $\overline{1} \in V_{n-1-k}$. Т.к. нам не важен порядок столбцов X_2^T и X_3^T, то количество разных X_2^T и X_3^T при фиксированном $\widetilde{x}_1^T$ с k нулями равно $(2^{n-k-1} - 2)/2 = 2^{n-k-2} - 1$.

Таким образом, мы получили следующие соотношения:

$$\#\{I_L \in \mathrm{BCI}(n, 4) \mid L < V_n\} = \sum_{k=0}^{n-2} \binom{n-1}{k} \cdot 2^{n-k-2} - \sum_{k=0}^{n-2} \binom{n-1}{k} =$$

$$= \frac{1}{2}\left(\sum_{k=0}^{n-1} \binom{n-1}{k} \cdot 2^{n-1-k} - 1\right) - 2^{n-1} + 1 =$$

$$= \frac{1}{2}(3^{n-1} - 1) - 2^{n-1} + 1 = \frac{1}{2}(3^{n-1} - 2^n + 1).$$

Для того, чтобы посчитать $\mathrm{BCI}(n, 4)$, нужно также учесть функции, которые получаются из подпространств путем сдвига на некоторые векторы. Т.к. для $L_i \neq L_j$ и $\forall\, x, y \in V_n$ $L_i \oplus x \neq L_j \oplus y$, то

$$\#\,\mathrm{BCI}(n, 4) = 2^{n-2} \cdot \#\{I_L \in \mathrm{BCI}(n, 4) \mid L < V_n\} = 2^{n-3}(3^{n-1} - 2^n + 1).$$

Что и требовалось. □

Следствие 1. $\#\{f \in \mathrm{CI}(n) \mid \mathrm{wt}(f) = 4\} = 2^{n-3}(3^{n-1} - 2^n + 1) + \binom{2^{n-1}}{2}$.

Доказательство. Непосредственно следует из предложения 6 и равенства $\mathrm{CI}(n) = \bigcup_{g \in \mathrm{BCI}(n)} (g \oplus \mathrm{Mir}(n)|_g)$. □

6. Пространства корреляционно-иммунных функций

Пусть $L < V_n$ такова, что $I_L \in \mathrm{CI}(n)$. Рассмотрим всевозможные различные сдвиги L на $z \in V_n$. Все пространство V_n состоит из $2^{n-\dim L}$ различных сдвигов пространства L на векторы из V_n.

Пусть $P_L = \{L, L \oplus z_1, \ldots, L \oplus z_{2^{n-\dim L}-1}\}$. Если $P_L^i = L \oplus z_i$, то $\forall\, i, j$: $i \neq j$ выполнено $P_L^i \cap P_L^j = \varnothing$ и $I_{P_L^i} \in \mathrm{CI}(n)$. Следовательно,

$$Z\left(\left\{I_{P_L^i}\right\}_{i=0}^{2^{n-\dim L}}\right) \subset \mathrm{CI}(n),$$

где $Z(S)$ означает линейную оболочку, натянутую на векторы из множества S.

Пусть T_n — множество сдвигов на векторы из V_n, а $J_{T_n}(f)$ — группа инерции функции f в группе T_n.

Предложение 7. *Пусть $f \in \mathcal{F}_n$. Если $J_{T_n}(f) \geqslant L$, то функция f представима в виде*

$$f = \sum_{i=1}^{2^{n-\dim L}} \varepsilon_i I_{L+z_i},$$

где $\varepsilon_i \in \{0, 1\}$, $z_i \in V_n$. То есть $f \in Z\left(\left\{P_L^i\right\}_{i=0}^{2^{n-\dim L}}\right)$.

Доказательство. Из условия видно, что $1_f \oplus v = 1_f$ для любого $v \in L$. Таким образом, для любого $z \in 1_f$ и любого $v \in L$ существует такой вектор $y \in 1_f$, что $z \oplus v = y$. Рассмотрим произвольный вектор $z_1 \in 1_f$. Множество $L \oplus z_1$ содержится в 1_f. Далее рассмотрим любой вектор z_2 из $1_f \setminus (L \oplus z_1)$. Множество $L \oplus z_2$ содержится в $1_f \setminus (L \oplus z_1)$. Продолжая далее рассмотрение векторов z_i, определяемых по описанной схеме, получим, что множество 1_f исчерпывается некоторым количеством сдвигов плоскости L. □

Признаком принадлежности произвольной булевой функции классу LCI(n) может служить свойство существования такого подпространства $L < V_n$, что $L \leqslant J_{T_n}(f)$ и $d(L^\perp) > 1$.

Введем обозначение $Z(L) = Z\left(\left\{P_L^i\right\}_{i=0}^{2^{n-\dim L}}\right)$. Понятно, что $\#Z(L) = 2^{2^{n-\dim L}}$. Причем если $L < V_n$ таково, что I_L — минимальная корреляционно-иммунная функция в классе LCI(n), то нельзя указать другое подпространство $S < V_n$ такое, что $Z(S) \supset Z(L)$.

Литература

[1] *Логачёв О. А.*, *Сальников А. А.*, *Ященко В. В.* Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2004.

[2] *Алексеев Е. К.* О некоторых алгебраических и комбинаторных свойствах множества корреляционно-иммунных функций в целом // Материалы IX Международного семинара «Дискретная математика и ее приложения». М.: Изд-во механико-математического факультета МГУ, 2007, 477 с.

[3] *Hedayat A. S.*, *Sloane N. J. A.*, *Stufken J.* Orthogonal arrays: theory and applications. Springer-Verlag, 1999.

Об использовании аффинных нормальных форм булевых функций для определения ключей фильтрующих генераторов

О. А. Логачёв

Аннотация

В работе предложен новый класс методов криптоанализа фильтрующих генераторов, использующий свойства аффинных нормальных форм фильтрующих булевых функций.

Ключевые слова: булева функция, фильтрующий генератор, аффинная нормальная форма, разбиение пространства, локальная аффинность, аффинная траектория.

1. Введение

Решение систем булевых уравнений, описывающих функционирование фильтрующих генераторов, является активно исследуемым направлением в криптологии. В ряде случаев для нахождения решения таких систем используются теоретико-вероятностные, статистические и теоретико-кодовые методы. В ряде случаев исходная истема погружается в действительную область, и решение находится с помощью соответствующих псевдобулевых неравенств. Известны алгебраические методы решения таких систем, использующие базисы Грёбнера. Наиболее эффективными методами являются методы линеаризации (алгебраическая и быстрая алгебраическая атаки). В последнее время активно исследуются возможности использования для решения этих систем решателей задачи выполнимости.

В настоящей статье предлагается метод решения указанных систем булевых уравнений, основанный на использовании аффинной нормальной формы фильтрующей функции. Использование аффинных ограничений фильтрующих функций позволяет использовать полиномиальные алгоритмы для нахождения ключа фильтрующего генератора.

2. Основные понятия, определения и обозначения

Пусть $\mathbb{F}_2$ — поле Галуа из двух элементов, $\mathbb{F}_2^n$ — векторное пространство наборов-строк с n (n — натуральное) компонентами из $\mathbb{F}_2$. Число единиц в наборе $x \in \mathbb{F}_2^n$ называют весом Хэмминга этого набора и обозначают $\mathrm{wt}(x)$. Булевой функцией от n переменных будем называть отображение из $\mathbb{F}_2^n$ в $\mathbb{F}_2$. Через $\oplus$ будем обозначать сложение в $\mathbb{F}_2$ и покомпонентное сложение в $\mathbb{F}_2^n$. Множество всех булевых функций от n переменных будем обозначать $\mathcal{F}_n$. Алгебраической нормальной формой (а. н. ф.) булевой функции называется многочлен из кольца $R_n = \mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 \oplus x_1, \ldots, x_n^2 \oplus x_n)$, однозначно представляющий эту функцию. Для f из $\mathcal{F}_n$ ее а. н. ф. будем обозначать

$$f(x) = \bigoplus_{u \in \mathbb{F}_2^n} a(u) x^u,$$

где $a(u) \in \mathbb{F}_2$, $x^u = x_1^{u_1} \cdot \ldots \cdot x_n^{u_n}$. Для монома x^u его длиной называют $\mathrm{wt}(u)$. Алгебраической степенью функции f называют величину

$$\deg(f) = \max\{\mathrm{wt}(a(u)) \mid a(u) = 1\}.$$

Будем обозначать через $\mathcal{A}_n$ подмножество аффинных булевых функций, т. е. булевых функций, алгебраическая степень которых не превосходит 1. Мощность произвольного конечного множества E будем обозначать $\mathrm{card}\, E$.

Пусть L — произвольное подпространство пространства $\mathbb{F}_2^n$ и $u \in \mathbb{F}_2$. Смежный класс $\pi = u \oplus L$ будем называть плоскостью в пространстве $\mathbb{F}_2^n$ и будем считать, что $\dim \pi = \dim L$. Множество всех плоскостей пространства $\mathbb{F}_2^n$ будем обозначать $\mathcal{P}(\mathbb{F}_2^n)$. Для плоскости π через I_π будем обозначать булеву функцию-индикатор этой плоскости, т. е.

$$I_\pi(x) = \begin{cases} 1, & \text{если } x \in \pi; \\ 0, & \text{если } x \notin \pi. \end{cases}$$

Хорошо известно (см., например, [1]), что I_π может быть представлена в виде

$$I_\pi(x) = \prod_{i=1}^{r} [\langle v^{(i)}, x \rangle \oplus \delta_i \oplus 1],$$

где $\langle a, b \rangle = a_1 b_1 \oplus \ldots \oplus a_n b_n$ — скалярное произведение строк a и b; $\delta_i \in \mathbb{F}_2$; $\mathrm{rank}\, v^{(i)} = r = n - \dim \pi_i = n - \dim L$ $(i = 1, 2, \ldots, r)$.

Пусть $\pi \in \mathcal{P}(\mathbb{F}_2^n)$. Для булевой функции $f \in \mathcal{F}_n$ через $f|_\pi = f'$ будем обозначать ограничение функции f на плоскость π, т. е. $f' : \pi \to \mathbb{F}_2$.

Локальной аффинностью (см., например, [2]) булевой функции f из $\mathcal{F}_n$ будем называть плоскость $\pi \in \mathcal{P}(\mathbb{F}_2^n)$ такую, что для некоторой функции $l \in$

$\in \mathcal{A}_n$ выполняется соотношение $f|_\pi = l|_\pi$. Через $\mathcal{P}(f)$ будем обозначать совокупность локальных аффинностей функции f. Множество $\mathcal{P}(f)$ является частично упорядоченным относительно теоретико-множественного включения. Обозначим через $p(f)$ максимальную размерность элементов из $\mathcal{P}(f)$.

Пусть $f \in \mathcal{F}_n$ и $\Pi = \{\pi_1, \ldots, \pi_s\}$ — разбиение пространства $\mathbb{F}_2^n$ на плоскости. Разбиение назовем аффинным разбиением для функции f, если любая плоскость из Π является локальной аффинностью для функции f. Совокупность аффинных разбиений для функции f будем обозначать $\Pi(f)$. Множество $\Pi(f)$ частично упорядочено относительно отношения $\preccurlyeq$ так, что $\Pi \preccurlyeq \Pi'$ тогда и только тогда, когда для любого $\pi \in \Pi$ существует $\pi' \in \Pi'$ такая, что $\pi \subseteq \pi'$ (другими словами, если $\Pi \neq \Pi'$, то Π' является укрупнением разбиения Π). Совокупность максимальных элементов множества $\Pi(f)$ будем обозначать $\Pi_{\max}(f)$, а $d_{\max}(f) = \max_{\Pi \in \Pi(f)} \max_{\pi \in \Pi} \dim \pi$.

3. Постановка задачи

Рассмотрим атаку по открытому и шифрованному текстам на потоковый шифр, построенный на основе фильтрующего генератора (т. е. регистра сдвига с линейными обратными связями и фильтрующей булевой функцией), с угрозой вскрытия ключа (см. рис. 1).

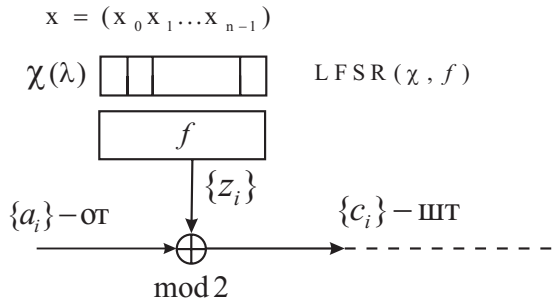


Рис. 1

Будем через LFSR(χ, f) обозначать фильтрующий генератор, построенный на основе двоичного регистра сдвига длины n с полиномом обратных связей $\chi(\lambda) = \chi_0 \oplus \chi_1\lambda \oplus \ldots \oplus \chi_n\lambda^n$ ($\chi_0 = \chi_n = 1$) и булевой фильтрующей функцией f из $\mathcal{F}_n$. Ключом LFSR(χ, f) является двоичная строка $x = (x_0x_1 \ldots x_{n-1})$, представляющая собой начальный отрезок линейной рекуррентной последовательности $\{x_i\}$. Последовательности $\{c_i\}$ и $\{a_i\}$ известны (т. е. известна последовательность $\{z_i\}$). Задача состоит

в восстановлении по известной последовательности $\{z_i\}$ длины N ключа $(x_0x_1 \dots x_{n-1})$, что равносильно восстановлению всей последовательности $\{x_i\}$. Матрица линейного преобразования, реализуемого регистром сдвига, имеет вид

$$\mathrm{L} = \begin{bmatrix} 0 & 0 & \dots & 0 & \chi_0 \\ 1 & 0 & \dots & 0 & \chi_1 \\ 0 & 1 & \dots & 0 & \chi_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & \chi_{n-2} \\ 0 & 0 & \dots & 1 & \chi_{n-1} \end{bmatrix}$$

Следовательно, криптографическая задача нахождения ключа эквивалентна математической задаче нахождения решения системы полиномиальных булевых уравнений вида

$$\begin{cases} f(x_0x_1 \dots x_{n-1}) = z_0, \\ f((x_0x_1 \dots x_{n-1})\mathrm{L}) = z_1, \\ \dots \\ f((x_0x_1 \dots x_{n-1})\mathrm{L}^r) = z_r, \\ \dots \\ f((x_0x_1 \dots x_{n-1})\mathrm{L}^{N-1}) = z_{N-1}. \end{cases} \tag{1}$$

относительно ключа $(x_0x_1 \dots x_{n-1})$.

4. Описание метода

Пусть $f \in \mathcal{F}_n$ и $\Pi = \{\pi_1, \pi_, \dots, \pi_s\} \in \Pi(f)$. Аффинной нормальной формой (аф. н. ф.) булевой функции f (см. [2]) будем называть выражение вида

$$f(x) = \bigoplus_{i=1}^{s} I_{\pi_i}(x) l_{\pi_i}(x), \tag{2}$$

где

$$I_{\pi_i}(x) = I_i(x) = \prod_{j=1}^{r_i} \left[\langle v^{ij}, x \rangle \oplus \delta_i \oplus 1 \right], \tag{3}$$

$r_i = n - \dim \pi_i$, $\delta_i \in \mathbb{F}_2$ и

$$f|_{\pi_i}(x) = l_{\pi_i}(x) = l_i(x) = \langle v^i, x \rangle \oplus \alpha_i, \tag{4}$$

$\alpha_i \in \mathbb{F}_2$, $i = 1, 2, \dots, s$.

Пусть $x \in \mathbb{F}_2^n$. Конечную последовательность векторов

$$(x,\ x\mathrm{L},\ \ldots,\ x\mathrm{L}^t,\ \ldots,\ x\mathrm{L}^{N-1})$$

будем называть траекторией вектора x относительно линейного преобразования L. Поскольку $\Pi = \{\pi_1, \pi_2, \ldots, \pi_s\} \in \Pi(f)$ является разбиением пространства $\mathbb{F}_2^n$, то

$$x \in \pi_{i_0},\ \ x\mathrm{L} \in \pi_{i_1},\ \ldots,\ \ x\mathrm{L}^t \in \pi_{i_t},\ \ldots,\ \ x\mathrm{L}^{N-1} \in \pi_{i_{N-1}}$$

для некоторого однозначно определенного набора индексов $i_0, i_1, \ldots, i_t, \ldots, i_{N-1}$, $1 \leqslant i_t \leqslant s$, $t = 0, 1, \ldots, N-1$.

Последовательность плоскостей

$$(\pi_{i_0}, \pi_{i_1}, \ldots, \pi_{i_t}, \ldots, \pi_{i_{N-1}}) \tag{5}$$

$\pi_{i_t} \in \Pi$, $t = 0, 1, \ldots, N-1$ будем называть аффинной траекторией набора x относительно преобразования L, задаваемой разбиением на локальные аффинности $\Pi \in \Pi(f)$.

Пусть нам известна последовательность

$$z = (z_0, z_1, \ldots, z_t, \ldots, z_{N-1}) \tag{6}$$

правых частей системы (1). Основная идея нахождения неизвестного набора $x = (x_0 x_1 \ldots x_{n-1})$ состоит в сведении решения нелинейной системы (1) к решению некоторой совокупности линейных систем, определяемой аф. н. ф. функции f, отображением L и последовательностью (6). Будем опробовать аффинные траектории вида (5). Всего таких траекторий будет s^N.

Предположение о том, что $x\mathrm{L}^t \in \pi_{i_t}$ влечет за собой выполнение равенства

$$f(x\mathrm{L}^t) = \bigoplus_{i=1}^{s} I_i(x\mathrm{L}^t) l_i(x\mathrm{L}^t) = I_{i_t}(x\mathrm{L}^t) l_{i_t}(x\mathrm{L}^t) = z_t,$$

т. е.

$$I_{i_t}(x\mathrm{L}^t) = 1, \quad l_{i_t}(x\mathrm{L}^t) = z_t.$$

Последние равенства эквивалентны системе линейных уравнений вида

$$\begin{cases} l_{i_t 1}(x\mathrm{L}^t) = l'_{i_t 1}(x) = 1, \\ \ldots \\ l_{i_t d_{i_t}}(x\mathrm{L}^t) = l'_{i_t d_{i_t}}(x) = 1, \\ l_{i_t}(x\mathrm{L}^t) = l'_{i_t}(x) = z_t. \end{cases} \tag{7}$$

Систему (7) можно привести к виду

$$\begin{cases} l'_{i_t1}(x) \oplus l'_{i_t1}(0) = 1 \oplus l'_{i_t1}(0) = a_{i_t1}, \\ \dots \\ l'_{i_td_{i_t}}(x) \oplus l'_{i_td_{i_t}}(0) = 1 \oplus l'_{i_td_{i_t}}(0) = a_{i_td_{i_t}}, \\ l'_{i_t}(x) \oplus l'_{i_t}(0) = z_t \oplus l'_{i_t}(0) = z_t^* \end{cases} \tag{8}$$

и представить в матричной форме

$$xA_{i_t} = (a_{i_t1} \dots a_{i_td_{i_t}} z_t^*) = \widetilde{z}_t, \tag{9}$$

где A_{i_t} — $n \times (d_{i_t} + 1)$-матрица. Предположение о том, что набор x имел аффинную траекторию (5), приводит нас к системе линейных уравнений

$$xB_N = x[A_{i_0}A_{i_1} \dots A_{i_t} \dots A_{i_{N-1}}] = (\widetilde{z}_0\widetilde{z}_1 \dots \widetilde{z}_t \dots \widetilde{z}_{N-1}), \tag{10}$$

где $B_N = [A_{i_0}A_{i_1} \dots A_{i_t} \dots A_{i_{N-1}}]$ — $n \times \left(\sum\limits_{k=0}^{N-1} d_{i_k} + N\right)$-матрица.

Этап 1.

1.1. Исходным элементом является разбиение на локальные аффинности $\Pi \in \Pi(f)$.

1.2. Из $\Pi \times \Pi$ выделяем $\mathcal{M}_2$ — множество аффинных траекторий длины 2, для которых выполняется равенство

$$\operatorname{rank} B_2 = \operatorname{rank} \begin{bmatrix} B_2 \\ \widetilde{z}_0\widetilde{z}_1 \end{bmatrix}.$$

1.3. Из $\mathcal{M}_2 \times \Pi$ выделяем $\mathcal{M}_3$ — множество аффинных траекторий длины 3, для которых выполняется равенство

$$\operatorname{rank} B_3 = \operatorname{rank} \begin{bmatrix} B_3 \\ \widetilde{z}_0\widetilde{z}_1\widetilde{z}_2 \end{bmatrix}.$$

…

1.t. Из $\mathcal{M}_{t-1} \times \Pi$ выделяем $\mathcal{M}_t$ — множество аффинных траекторий длины t, для которых выполняется равенство

$$\operatorname{rank} B_t = \operatorname{rank} \begin{bmatrix} B_t \\ \widetilde{z}_0\widetilde{z}_1 \dots \widetilde{z}_t \end{bmatrix}.$$

…

1.N. Из $\mathcal{M}_{N-1} \times \Pi$ выделяем $\mathcal{M}_N$ — множество аффинных траекторий длины N, для которых выполняется равенство

$$\operatorname{rank} B_N = \operatorname{rank} \begin{bmatrix} B_N \\ \widetilde{z}_0\widetilde{z}_1 \dots \widetilde{z}_{N-1} \end{bmatrix}.$$

Этап 2. Для аффинных траекторий из множества $\mathcal{M}_N$ находим решения соответствующих линейных систем.

Этап 3. Проверяем все полученные на предыдущем этапе возможные начальные состояния фильтрующего генератора на соотвествие выходной последовательности длины большей n (т. е. для решения данной задачи может потребоваться выходная последовательность длины большей чем расстояние единственности). После отбраковки ложных начальных состояний находим начальное заполнение.

При расчетах трудоемкости этапа 1 метода можно в предварительном порядке использовать особенности аф. н. ф. функции f вне зависимости от последовательности z. Пусть $(\pi_i, \pi_j) \in \Pi \times \Pi$. Обозначим через C_{i_t} матрицу первых d_{i_t} уравнений системы (7). Рассмотрим линейную систему вида

$$x[C_iC_j] = (a_{i_1} \dots a_{i_{d_i}} a_{j_1} \dots a_{j_{d_j}}). \tag{11}$$

Если

$$\operatorname{rank}[C_iC_j] \neq \operatorname{rank} \begin{bmatrix} C_iC_j \\ a_{i_1} \dots a_{i_{d_i}} a_{j_1} \dots a_{j_{d_j}} \end{bmatrix}, \tag{12}$$

то локальные аффинности π_i и π_j будем называть противоречивыми, а в противном случае — согласованными. Очевидно, что если система (11) не имеет решения, т. е. выполнено (12), то и система

$$x[A_iA_j] = (a_{i_1} \dots a_{i_{d_i}} \alpha a_{j_1} \dots a_{j_{d_j}} \beta)$$

не имеет решения при любых $\alpha, \beta \in \mathbb{F}_2$. Обозначим через $\mathcal{M}_1$ множество непротиворечивых пар локальных аффинностей из Π. Это множество может быть использовано на первом этапе метода для сокращения трудоемкости. На этапе 1 для расчета параметров метода можно использовать предположение о том, что строки матриц, описывающих системы линейных уравнений, распределены равномерно и независимо в соотвествующем пространстве. На этапах 1 и 2 используются полиномиальные алгоритмы нахождения ранга матрицы и решения систем линейных уравнений. Последовательность $\{\operatorname{card} \mathcal{M}_t \mid t = 2, 3, \dots, N\}$, определяющая эффективность метода, существенно зависит от параметров

$$n,\ N,\ \chi,\ \{d_i \mid i = 1, \dots, s\},\ \{A_i \mid i = 1, \dots, s\},\ (z_0, z_1, \dots, z_{N-1}).$$

В заключение приведем пример фильтрующей булевой функции потокового шифра LILI-128 [3], а также аф. н. ф. этой функции, найден-

ную в [4]:

$$f(x_1, x_2, \ldots, x_{10}) = x_5 \oplus x_4 \oplus x_3 \oplus x_2 \oplus x_{10}x_6 \oplus x_{10}x_4 \oplus x_9x_3 \oplus x_9x_1 \oplus x_8x_2 \oplus$$
$$\oplus x_8x_1 \oplus x_7x_6 \oplus x_{10}x_9x_5 \oplus x_{10}x_9x_4 \oplus x_{10}x_9x_3 \oplus x_{10}x_9x_2 \oplus x_{10}x_8x_4 \oplus$$
$$\oplus x_{10}x_8x_3 \oplus x_{10}x_7x_6 \oplus x_{10}x_7x_5 \oplus x_{10}x_7x_4 \oplus x_9x_8x_6 \oplus x_9x_8x_3 \oplus x_9x_7x_6 \oplus$$
$$\oplus x_9x_7x_4 \oplus x_9x_7x_3 \oplus x_{10}x_9x_8x_6 \oplus x_{10}x_9x_8x_4 \oplus x_{10}x_9x_8x_3 \oplus x_{10}x_9x_8x_1 \oplus$$
$$\oplus x_{10}x_9x_7x_6 \oplus x_{10}x_9x_7x_4 \oplus x_{10}x_9x_7x_2 \oplus x_{10}x_8x_7x_5 \oplus x_{10}x_8x_7x_3 \oplus$$
$$\oplus x_9x_8x_7x_4 \oplus x_9x_8x_7x_2 \oplus x_9x_7x_6x_5 \oplus x_9x_7x_6x_4 \oplus x_{10}x_9x_8x_7x_4 \oplus$$
$$\oplus x_{10}x_9x_8x_7x_3 \oplus x_{10}x_9x_7x_6x_5 \oplus x_{10}x_9x_7x_6x_4 \oplus x_9x_8x_7x_6x_5 \oplus$$
$$\oplus x_9x_8x_7x_6x_4 \oplus x_{10}x_9x_8x_7x_6x_5 \oplus x_{10}x_9x_8x_7x_6x_4.$$

Аф. н. ф. имеет вид

$$f(x) = \bigoplus_{i=1}^{33} I_i(x) l_i(x),$$

где I_i и l_i задаются таблицами 1 и 2.

Таблица 1. Индикаторные функции аффинной нормальной формы функции потокового шифра LILI-128

i	I_i	i	I_i
1	$\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4$	18	$\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4(x_7 \oplus x_5)$
2	$\overline{x}_1x_2\overline{x}_3\overline{x}_4\overline{x}_5$	19	$x_1\overline{x}_2\overline{x}_3\overline{x}_4(x_7 \oplus x_5)$
3	$x_1x_2\overline{x}_3\overline{x}_4\overline{x}_5$	20	$\overline{x}_1x_2\overline{x}_3\overline{x}_4(x_7 \oplus x_5)$
4	$\overline{x}_1\overline{x}_2x_3\overline{x}_4(x_{10} \oplus x_9 \oplus 1)$	21	$x_1x_2\overline{x}_3\overline{x}_4(x_7 \oplus x_5)$
5	$x_1\overline{x}_2x_3\overline{x}_4(x_{10} \oplus x_9 \oplus 1)$	22	$\overline{x}_1\overline{x}_2x_3\overline{x}_4(x_{10} \oplus x_9 \oplus x_8 \oplus x_5)$
6	$\overline{x}_1x_2x_3\overline{x}_4(x_{10} \oplus x_9 \oplus x_5 \oplus 1)$	23	$x_1\overline{x}_2x_3\overline{x}_4(x_{10} \oplus x_9 \oplus x_8 \oplus x_5)$
7	$x_1x_2x_3\overline{x}_4(x_{10} \oplus x_9 \oplus x_5 \oplus 1)$	24	$\overline{x}_1x_2x_3\overline{x}_4(x_{10} \oplus x_9 \oplus x_7 \oplus x_5)$
8	$\overline{x}_1\overline{x}_2\overline{x}_3x_4(x_{10} \oplus x_8 \oplus 1)$	25	$x_1x_2x_3\overline{x}_4(x_{10} \oplus x_9 \oplus x_7 \oplus x_5)$
9	$x_1\overline{x}_2\overline{x}_3x_4(x_{10} \oplus x_8 \oplus 1)$	26	$\overline{x}_1\overline{x}_2\overline{x}_3x_4(x_{10} \oplus x_9 \oplus x_6 \oplus x_5)$
10	$\overline{x}_1\overline{x}_2x_3\overline{x}_4x_5(x_{10} \oplus x_7 \oplus 1)$	27	$x_1\overline{x}_2\overline{x}_3x_4(x_{10} \oplus x_9 \oplus x_6 \oplus x_5)$
11	$x_1\overline{x}_2x_3\overline{x}_4x_5(x_{10} \oplus x_7 \oplus 1)$	28	$\overline{x}_1x_2\overline{x}_3x_4(x_{10} \oplus x_8 \oplus x_7 \oplus x_5)$
12	$\overline{x}_1x_2x_3\overline{x}_4x_5(x_{10} \oplus x_6 \oplus 1)$	29	$x_1x_2\overline{x}_3x_4(x_{10} \oplus x_8 \oplus x_7 \oplus x_5)$
13	$x_1x_2x_3\overline{x}_4x_5(x_{10} \oplus x_6 \oplus 1)$	30	$\overline{x}_1\overline{x}_2x_3x_4(x_{10} \oplus x_8 \oplus x_6 \oplus x_5)$
14	$\overline{x}_1\overline{x}_2x_3x_4(x_9 \oplus x_5 \oplus 1)$	31	$x_1\overline{x}_2x_3x_4(x_{10} \oplus x_8 \oplus x_6 \oplus x_5)$
15	$x_1\overline{x}_2x_3x_4(x_9 \oplus x_5 \oplus 1)$	32	$\overline{x}_1x_2x_3x_4(x_{10} \oplus x_7 \oplus x_6 \oplus x_5)$
16	$x_1x_2x_3x_4(x_8 \oplus x_5 \oplus 1)$	33	$x_1x_2x_3x_4(x_{10} \oplus x_7 \oplus x_6 \oplus x_5)$
17	$\overline{x}_1x_2x_3x_4(x_8 \oplus x_5 \oplus 1)$		

Таблица 2. Аффинные функции, представляющие фильтрующую функцию потокового шифра LILI-128 на локальных аффинностях, определеяемых табл. 1

i	l_i	i	l_i
1	$x_{10}x_9 \oplus x_8 \oplus x_7$	18	$x_{10} \oplus x_9 \oplus x_8$
2	$x_{10}x_9 \oplus x_8 \oplus x_7$	19	$x_{10} \oplus x_9 \oplus x_7$
3	$x_{10}x_9 \oplus x_8 \oplus x_7 \oplus 1$	20	$x_{10} \oplus x_9 \oplus x_8$
4	$x_9 \oplus x_8 \oplus x_7$	21	$x_{10} \oplus x_9 \oplus x_8$
5	$x_9 \oplus x_8 \oplus x_7 \oplus 1$	22	$x_8 \oplus x_7$
6	$x_9 \oplus x_8 \oplus x_7$	23	$x_8 \oplus x_7 \oplus 1$
7	$x_9 \oplus x_8 \oplus x_7 \oplus 1$	24	$x_9 \oplus x_7$
8	$x_{10} \oplus x_8 \oplus x_7$	25	$x_9 \oplus x_7 \oplus 1$
9	$x_{10} \oplus x_8 \oplus x_7 \oplus 1$	26	$x_9 \oplus x_8$
10	$x_{10} \oplus x_9 \oplus x_7$	27	$x_9 \oplus x_8 \oplus 1$
11	$x_{10} \oplus x_9 \oplus x_7 \oplus 1$	28	$x_{10} \oplus x_7$
12	$x_{10} \oplus x_9 \oplus x_8$	29	$x_{10} \oplus x_7 \oplus 1$
13	$x_{10} \oplus x_9 \oplus x_8 \oplus 1$	30	$x_{10} \oplus x_8$
14	$x_9 \oplus x_8 \oplus x_7$	31	$x_{10} \oplus x_8 \oplus 1$
15	$x_9 \oplus x_8 \oplus x_7$	32	$x_{10} \oplus x_9$
16	$x_{10} \oplus x_8 \oplus x_7$	33	$x_{10} \oplus x_9 \oplus 1$
17	$x_{10} \oplus x_8 \oplus x_7$		

Литература

[1] *Мак-Вильямс Ф. Дж.*, *Слоэн Н. Дж. А.* Теория кодов, исправляющих ошибки. М.: Связь, 1979.

[2] *Logachev O. A.*, *Yashchenko V. V.*, *Denisenko M. P.* Local affinity of Boolean mappings // Proc. of the NATO Advanced Study Institute on Boolean Functions in Cryptology and Information Security (Zvenigorod, Moscow region, Russia, September 8–18, 2007). IOS Press, 2008, p. 148–172.

[3] *Huang X.*, *Huang W.*, *Liu X.*, *Wang C.*, *Wang Z. J.*, *Wang T.* Reconstructing the nonlinear filter function of LILI-128 stream cipher based on complexity. arXiv:cs/0702128 (`http://arxiv.org/`).

[4] *Гаврилушкин П. А.* Построение и анализ эффективности алгоритмов решения систем булевых уравнений, заданных аффинными нормальными формами. Дипломная работа, Факультет ВМК МГУ имени М. В. Ломоносова.

Часть II

СЕКЦИЯ «МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ КОМПЬЮТЕРНЫХ СИСТЕМ»

О восстановлении разбиения множества состояний безопасности

А. В. Галатенко

Аннотация

В работе рассматривается автоматная система, часть состояний которой объявляется безопасной. Исследуется сложность восстановления разбиения множества состояний для безопасных и ε-безопасных языков, введенных в работе «Автоматные модели защищенных компьютерных систем».

1. Основные понятия и результаты

Под конечным автоматом мы будем понимать четверку $V = (A, Q, \varphi, q_0)$, где A — конечное множество входных символов, Q — конечное множество состояний, $\varphi\colon A \times Q \to Q$ — функция переходов, $q_0 \in Q$ — начальное состояние. Пусть $Q = S \cup I$, причем $S \cap I = \varnothing$. Состояния из S назовем безопасными, состояния из I — небезопасными. Далее будем предполагать, что начальное состояние является безопасным, все состояния достижимы из начального, а $|Q| > 1$.

Обозначим через A^* множество всех конечных слов в алфавите A. Функция φ может быть продолжена на множество $A^* \times Q$ по мультипликативности.

Подмножество A^* называется языком. Каждому слову $\alpha \in A^*$ соответствует слово $\varkappa(\alpha) \in Q^*$, $\varkappa(\alpha) = \varphi(\alpha, q_0)$. Назовем слово $\alpha \in A^*$ безопасным, если $\varkappa(\alpha) \in S^*$. Назовем язык $\mathcal{A} \subseteq A^*$ безопасным (S-языком), если все слова, составляющие $\mathcal{A}$, безопасны, и не существует безопасных слов, не принадлежащих $\mathcal{A}$.

Решается следующая задача. Пусть известны A, Q, φ и q_0. Пусть имеется оракул $\psi\colon A^* \to \{0, 1\}$, для каждого входного слова $\alpha \in A^*$ выдающий 1 в том и только том случае, когда α безопасно. Так как все безопасные языки являются регулярными [1] и в силу теоремы Клини [2], $\psi(\alpha, q_0)$ может быть реализован автоматно. Для этого в качестве состояний следует рассматривать пары (q, ind), где $q \in Q$, а $\mathrm{ind} = 1$, если слово принадлежит

безопасному языку, и $\mathrm{ind} = 0$ в противном случае. В начальный момент времени $\mathrm{ind} = 1$. В момент времени $t+1$ $\mathrm{ind} = 0$ тогда и только тогда, когда либо $\mathrm{ind} = 0$ в момент времени t, либо когда первая компонента состояния в момент времени $t+1$ не принадлежит S. Добавив к автомату выходную функцию ψ, значения которой в момент t совпадают с ind в момент времени $t+1$. Для простоты изложения мы будем рассматривать только первую компоненту состояний и считать ψ оракулом. В этом случае можно считать, что функция выхода автомата V является индикатором того, что очередное состояние принадлежит S, а выходное слово дополнительно обрабатывается автоматом, выход которого принимает значение 1 тогда и только тогда, когда все входные буквы равнялись 1. Без ограничения общности будем считать, что все состояния из S достижимы из начального состояния по путям, содержащим только состояния из S. Требуется восстановить разбиение множества Q на S и I с помощью кратного условного эксперимента, подав минимальное количество входных слов или входные слова минимальной суммарной длины.

Рассмотрим эксперимент, в процессе которого на вход автомату V подается заданное множество входных слов, просматриваются значения функции выхода и восстанавливается разбиение Q на S и I. Обозначим через $N(V)$ наименьшее число подаваемых на вход слов, а через $NL(V)$ — наименьшую суммарную длину слов. Справедливо следующее утверждение.

Лемма 1. *$N(V) \geqslant 1$, $NL(V) \geqslant 1$, и эти оценки неулучшаемы.*

Определим функции Шеннона $L(n)$ и $LL(n)$ следующим образом: $L(n) = \max_{\{V:|Q|=n\}}(N(V))$, $LL(n) = \max_{\{V:|Q|=n\}}(NL(V))$.

Теорема 1. *Если мощность входного алфавита неограничена, то $L(n) = n-1$; если $|A| = k \geqslant 2$, то $L(n) = (n-1) - \left[\frac{n-2}{k}\right]$. Если мощность входного алфавита неограничена, то $LL(n) = \frac{n^2}{4}$ при четных n, $LL(n) = \frac{(n-1)^2}{4}$ при нечетных n; если мощность входного алфавита ограничена и больше или равна 2, то $\frac{n^2}{6} \leqslant LL(n) \leqslant \frac{n^2}{4}$.*

Напомним введенное в [1] понятие ε-безопасности. Рассмотрим произвольное $\varepsilon > 0$. Введем функции $s\colon Q^* \to \mathbb{N} \cup \{0\}$ и $i\colon Q^* \to \mathbb{N} \cup \{0\}$ следующим образом. Пусть $\varkappa \in Q^*$. Функция $s(\varkappa)$ равняется числу букв $\varkappa$, содержащихся в S, $i(\varkappa)$ равняется числу букв $\varkappa$, содержащихся в I. Обозначим через $|\varkappa|$ число букв в слове $\varkappa$. Назовем слово $\varkappa$ ε-безопасным, если $\frac{i(\varkappa)}{|\varkappa|} \leqslant \varepsilon$. Назовем язык $\mathcal{A}$ ε-безопасным (S_ε-языком), если все слова

из $\mathcal{A}$ ε-безопасны, и не существует ε-безопасных слов, не принадлежащих $\mathcal{A}$. Отметим, что можно рассматривать и $\varepsilon = 0$; в этом случае получим безопасные языки.

Рассмотрим задачу восстановления параметров ε-безопасности. Будем считать, что имеется оракул $\psi\colon A^* \to \{0, 1\}$, представляющий собой индикатор ε-безопасности, то есть принимающий значение 1 тогда и только тогда, когда $\dfrac{i(\varkappa)}{|\varkappa|} \leqslant \varepsilon$. Вообще говоря, оракул не является автоматным, например, в силу того, что ε-безопасные языки могут быть нерегулярными [1]. Пусть известно разбиение множества состояний Q на S и I, а значение ε неизвестно. Будем говорить, что ε_1 эквивалентно ε_2, если ε_1- и ε_2-безопасные языки совпадают. Легко увидеть, что введенное отношение действительно является эквивалентностью. Требуется восстановить значение ε с точностью до класса эквивалентности, подавая на вход автомату V конечное число слов из A^* и анализируя выход, или, другими словами, восстановить распознаваемый автоматом ε-безопасный язык.

Рассмотрим множество Δ значений $\dfrac{i(\varkappa)}{|\varkappa|}$ на всех словах из A^*. Пусть $\delta \in \mathbb{Q}$, $0 \leqslant \delta \leqslant 1$. Скажем, что δ принадлежит спектру автомата V, если δ является предельной точкой множества Δ, то есть в любой проколотой окрестности δ найдется хотя бы одна точка из Δ.

Спектр автомата может быть охарактеризован в терминах диаграммы Мура.

Теорема 2. *Пусть $\delta \in \mathbb{Q}$, $0 \leqslant \delta \leqslant 1$, V — автомат. Тогда δ принадлежит спектру V тогда и только тогда, когда выполнено хотя бы одно из следующих условий.*

1. *В диаграмме Мура V есть циклы C_1 и C_2, причем доля небезопасных состояний C_1 меньше δ, доля небезопасных состояний C_2 больше δ, и существует путь из C_1 в C_2 или из C_2 в C_1.*
2. *В диаграмме Мура V есть цикл C, в котором доля небезопасных состояний равна δ, $0 < \delta < 1$.*
3. *$\delta = 0$, в диаграмме Мура V есть цикл C, в котором все состояния безопасны, и существует путь, соединяющий C с некоторым небезопасным состоянием.*
4. *$\delta = 1$, в диаграмме Мура V есть цикл C, в котором все состояния небезопасны, и существует путь, соединяющий C с некоторым безопасным состоянием, при этом безопасное состояние не является первым в этом пути.*

Следствие 1. *Существует алгоритм, по диаграмме Мура автомата и разбиению множества состояний строящий спектр.*

Следствие 2. *Спектр любого автомата непуст тогда и только тогда, когда в диаграмме Мура существует путь, начинающийся с начального состояния и содержащий по крайней мере одну небезопасную вершину и одну безопасную вершину, не являющуюся первым элементом пути.*

Теорема 3. *Задача восстановления параметров* ε*-безопасности с помощью конечного эксперимента неразрешима, если одновременно не выполнены следующие два условия*:

1) ε *не принадлежит спектру автомата* V;

2) $\varepsilon = 1$, *и не существует перехода из начального состояния в небезопасное состояние по слову длины* 1.

Задача разрешима с помощью конечного эксперимента, если выполнено хотя бы одно из следующих условий:

1) *задано* $\nu > 0$, *такое что* ε *удалено от спектра автомата* V *не менее, чем на* ν;

2) $\varepsilon = 1$, *и не существует перехода из начального состояния в небезопасное состояние по слову длины* 1.

2. Доказательства утверждений

1. Доказательство леммы 1

Для восстановления множества S необходимо проверить, есть ли переходы из начального состояния в состояния множества S. Для этого на вход необходимо подать по крайней мере одну букву, следовательно $N(V) \geqslant 1$, $NL(V) \geqslant 1$. Покажем, что оценка неулучшаема. Рассмотрим автомат с диаграммой Мура, изображенной на рис. 1. Пусть множество S состоит только из начального состояния q_0. Чтобы восстановить такое разбиение, на вход достаточно подать произвольную букву. Лемма доказана. □

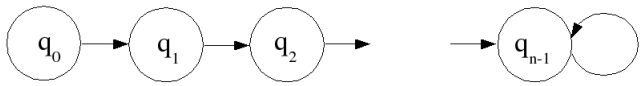


Рис. 1. Диаграмма автомата, на котором достигается нижняя оценка в лемме 1

2. Вспомогательные утверждения

Лемма 2 (о максимальном числе листьев). *Рассмотрим дерево с корнем, содержащее n вершин, $n > 1$. Если степень вершин неограничена, дерево содержит не более $n - 1$ листьев (корень не считается листом), и эта оценка достижима. Если степень вершин ограничена константой $k \in \mathbb{N}$, дерево содержит не более $(n-1) - \left[\frac{n-2}{k}\right]$ листьев, и эта оценка достижима.*

Доказательство. Пусть степень вершин неограничена. Так как корень дерева не считается листом, число листьев не превосходит общего числа вершин минус 1. Оценка достигается на дереве из двух ярусов (рис. 2).

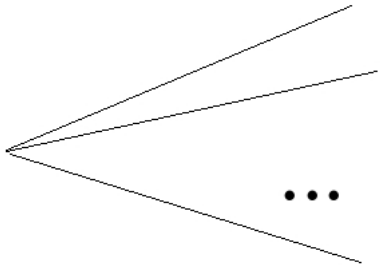


Рис. 2. Дерево из двух ярусов

Пусть степень вершин ограничена константой k. Покажем, что в этом случае максимальное число листьев имеет равномерно загруженное дерево, определяемое следующим образом:

1) все вершины, кроме, может быть, вершин предпоследнего яруса, имеют степень k;
2) На предпоследнем ярусе все вершины, кроме, может быть, одной, имеют степень k.

Действительно, рассмотрим произвольное дерево T, не являющееся равномерно загруженным. Если не выполнено условие 1, построим дерево T_1 следующим образом. Рассмотрим недогруженную вершину. Если она является листом, перевесим в нее поддерево с предпоследнего яруса. При этом число листьев не изменится. Если вершина не является листом, перевесим произвольное ребро с предпоследнего яруса. При этом число листьев не уменьшится. Будем последовательно строить деревья T_i, пока все вершины нижних ярусов не окажутся полностью загруженными. На

последнем этапе будем последовательно перевешивать ребра на предпоследнем ярусе, догружая недогруженные вершины. При этом число листьев не будет уменьшаться, так как при каждом таком преобразовании удаляется один лист, а добавляется — один или два. Учитывая, что на каждом шаге число листьев не уменьшалось, а на выходе получилось равномерно загруженное дерево, получаем искомое утверждение.

Покажем по индукции, что число $T(n)$ листьев равномерно загруженного дерева с n вершинами равно $(n-1) - \left[\frac{n-2}{k}\right]$. При $n = 2$ равенство очевидно. Пусть утверждение истинно для всех n, не превосходящих некоторого $N \in \mathbb{N}$. Покажем, что равенство справедливо при $n = N + 1$. Возможны два случая. Если одна из вершин предпоследнего яруса равномерно загруженного дерева с N вершинами недогружена, то $T(N+1) = T(N) + 1$, в противном случае $T(N+1) = T(N)$. Легко увидеть, что второму случаю соответствуют значения N вида $1 + C \times k$ для некоторого натурального C. В первом случае

$$T(N+1) = T(N) + 1 = (N-1) - \left[\frac{N-2}{k}\right] + 1 = ((N+1)-1) - \left[\frac{N-1}{k}\right],$$

так как N отличен от 1 по модулю k, а целая часть увеличивается, когда $N - 1$ кратно k. Во втором случае

$$T(N+1) = T(N) = (N-1) - \left[\frac{N-2}{k}\right] = (N+1-1) - \left[\frac{N-2}{k} + 1\right].$$

Так как $N = 1 + C \times k$, $\left[\frac{N-2}{k} + 1\right] = \left[\frac{N-1}{k}\right]$. □

Рассмотрим дерево T с корнем с n вершинами. Обозначим через $C(T)$ суммарную длину всех цепей, начинающихся от корня и заканчивающихся в листьях.

Лемма 3 (о длине цепей). *Для любого n существует бинарное дерево с корнем и n вершинами, для которого $C(T) > \frac{n^2}{6}$.*

Доказательство. Рассмотрим класс деревьев, изображенный на рис. 3. Пусть $n - l$ четно. Для дерева $T(n)$ с n вершинами имеем:

$$C(T(n)) = (l+1) + (l+2) + \ldots + \left(l + \left(\frac{n-l}{2}\right)\right) =$$

$$= \frac{n-l}{4} \times \left(2l + \frac{n-l}{2} + 1\right) = -\frac{3}{8}\left(l^2 - 2l\left(\frac{n-1}{3}\right) - \frac{n^2+2n}{3}\right).$$

Рассмотрим значение l, при котором $C(T(n))$ максимально. Так как выражение представляет собой квадратичную по l функцию с отрицательным старшим коэффициентом, максимум достигается при $l = \frac{n-1}{3}$. Учитывая

симметрию, монотонность левой ветви параболы, тот факт, что ровно одно из чисел $\left\{\frac{n-2}{3}, \frac{n-1}{3}, \frac{n}{3}\right\}$ является целым, и соображение, что для обеспечения четности $n - l$ из l, возможно, придется вычесть еще 1, получаем, что максимальное значение $C(T(n))$ не меньше значения, получаемого подстановкой вместо l значения $\frac{n}{3} - 1$, то есть $C((T(n)) >= \frac{n^2}{6} + \frac{n}{6} + \frac{9}{12} > \frac{n^2}{6}$. Последнее неравенство доказывает лемму. □

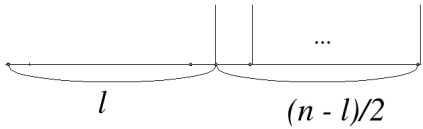


Рис. 3. Класс деревьев для доказательства леммы о длине цепей

Лемма 4 (о максимальных цепях). *Рассмотрим дерево с n вершинами. Суммарная длина простых цепей от корня к листьям не превосходит* $\frac{n^2}{4}$ *при четных n и* $\frac{(n-1)^2}{4}$ *— при нечетных n, при этом обе оценки достигаются.*

Доказательство. Покажем, что достаточно рассматривать деревья, изображенные на рис. 4. Действительно, рассмотрим произвольное дерево с n вершинами. Пусть k — максимальная длина простой цепи от корня к листьям. Обозначим цепь максимальной длины через γ. Будем последовательно выполнять следующее преобразование. Будем перевешивать листья, не прикрепленные к предпоследней вершине γ N, к N. В силу того, что длина γ максимальная, в результате суммарная длина цепей не уменьшится, так как в результате преобразования исчезнет цепь длины, не превос

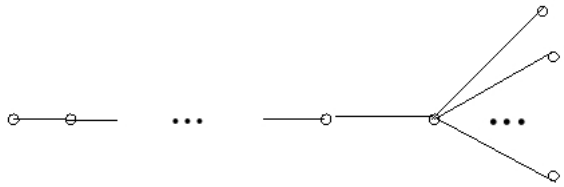


Рис. 4. Класс деревьев для доказательства леммы о максимальных цепях

Посчитаем суммарную длину простых цепей в получившемся дереве. Во всех ярусах, исключая последний, оказывается k вершин, следователь-

но, в дереве имеется $n-k$ листьев, а суммарная длина равна $k(n-k)$. Максимум достигается при $k=\dfrac{n}{2}$ и равен $\dfrac{n^2}{4}$. Если n нечетно, максимум достигается при $k=\dfrac{n-1}{2}$ и равен $\dfrac{(n-1)^2}{4}$. Так как оценки построены конструктивно, они достигаются на деревьях из класса на рис. 4 с соответствующими значениями k. Лемма доказана. □

Лемма 5 (о склеивании значений ε). *Пусть* ε_1, $\varepsilon_2 \in \mathbb{Q}$, $0 \leqslant \varepsilon_1 < \varepsilon_2 < 1$, *таковы, что для некоторого натурального* n *не существует таких* p, $q \in \mathbb{N}$, p, $q \leqslant n$, *что* $\varepsilon_1 \leqslant \dfrac{p}{q} < \varepsilon_2$. *Тогда* ε_1 *и* ε_2 *безопасные подъязыки, содержащие слова длины не более* n, *совпадают для любого автомата* V *и любого разбиения.*

Доказательство. Предположим противное. Пусть существует автомат V и разбиение множества его состояний на S и I, при котором существует слово α длины, не превосходящей n, являющееся ε_2-безопасным и не являющееся ε_1-безопасным (обратный случай, очевидно, невозможен). Пусть длина α равна q, число состояний из множества I равно p. Из построения ясно, что p, $q \leqslant n$. Из ε_2-безопасности α следует, что $\dfrac{p}{q} > \varepsilon_2$, из ε_1-небезопасности — что $\dfrac{p}{q} \leqslant \varepsilon_1$. Полученное противоречие доказывает лемму. □

3. Доказательство теоремы 1

Построим эксперимент следующим образом. Будем подавать на вход слова по одной букве. Подача слова заканчивается, если либо автомат переводится в вершину из множества I (то есть на выход выдается символ 0), либо из состояния, в которое перешел автомат, невозможны переходы в состояния, в которые автомат еще не попадал. Слова подаются, пока есть возможность достичь еще не рассмотренной вершины из множества S, то есть существуют пути в диаграмме Мура, начинающиеся в начальном состоянии, заканчивающиеся в еще не рассмотренном состоянии и содержащие только вершины, про которые либо известно, что они входят в S, либо ничего не известно. Легко увидеть, что в результате в диаграмме Мура автомата будет построено поддерево со следующими свойствами:

1) все вершины из множества S входят в дерево (это следует из достижимости всех состояний множества S из начального по путям, содержащим только состояния из S);
2) вершины из множества I могут входить только в качестве листьев (это следует из построения эксперимента).

Первое свойство гарантирует, что разбиение множества Q будет восстановлено правильно.

В силу леммы о максимальном числе листьев, общее число поданных слов для автомата с n состояниями не превосходит $n-1$, если входной алфавит неограничен, и $(n-1)-\left[\frac{n-2}{k}\right]$ — если мощность входного алфавита равна k. Нижняя оценка $L(n)$ следует из рассмотрения автомата, диаграмма Мура которого получается из построенного в доказательстве леммы о максимальном числе листьев в дереве приписыванием произвольных входных символов имеющимся ребрам с условием, что функция переходов сохранит однозначность определения, и добавлением недостающих переходов в виде петель. При этом $S=Q$, то есть дерево включит в себя все n состояний.

В силу леммы о максимальных цепях, верхняя оценка $LL(n)$ справедлива. Нижняя оценка получается из построенных в леммах о максимальных цепях и о длине цепей классов деревьев аналогично оценке $L(n)$. Теорема доказана. □

4. Доказательство теоремы 2

Покажем, что если выполнено условие 1, то $\delta=\frac{p}{q}$ принадлежит спектру. Без ограничения общности, существует путь из C_1 в C_2, и неравенство для C_1 является строгим. Пусть доля небезопасных состояний в C_1 равна $\frac{p_1}{q_1}$, где q_1 — длина C_1, доля небезопасных состояний в C_2 равна $\frac{p_2}{q_2}$, где q_2 — длина C_2, $\frac{p'}{q'}$ и $\frac{p''}{q''}$ — доли небезопасных состояний в пути, ведущем в C_1 и в пути из C_1 в C_2 соответственно. Рассмотрим последовательность, состоящую из пути в C_1, l_1 обходов C_1, пути в C_2 и l_2 обходов C_2. Рассмотрим последовательность значений

$$r_{l_1,l_2}=\frac{i(\varkappa)}{|\varkappa|}=\frac{p'+p''+l_1\times p_1+p_2\times l_2}{q'+q''+l_1\times q_1+l_2\times q_2}.$$

Рассмотрим произвольное $\gamma>0$. Так как $\frac{p_1}{q_1}<\frac{p}{q}$, а $\lim\limits_{l_1\to\infty} r_{l_1,1}=\frac{p_1}{q_1}$, то существует $L_1\in\mathbb{N}$, что $r_{l_1,1}<\frac{p}{q}$ для всех $l_1\geqslant L_1$. Рассмотрим разность $r_{l_1,l_2+1}-r_{l_1,l_2}$ при $l_1\geqslant L_1$. Легко увидеть, что в этом случае при фиксированном l_1 r_{l_1,l_2} монотонно возрастает по l_2. Действительно, r_{l_1,l_2} можно записать в виде $\frac{a+p_2\times l_2}{b+q_2\times l_2}$. Функция дифференцируема на $\mathbb{R}^+$, и знак производной постоянный, следовательно функция монотонна. При $l_2=1$ значение по построению меньше $\frac{p}{q}$, при $l_2\to\infty$ значение стремится к $\frac{p_2}{q_2}>\frac{p}{q}$,

следовательно функция возрастает, и рассматриваемая разность положительна. Оценим ее, заменив в знаменателе первой дроби $l_2 + 1$ на l_2:

$$r_{l_1,l_2+1} - r_{l_1,l_2} \geqslant \frac{p_2}{q' + q'' + l_1 \times q_1 + l_2 \times q_2}.$$

Выберем L'_1 таким образом, чтобы дробь в правой части оказалась меньше $\frac{\gamma}{2}$ при всех $l_1 \geqslant L'_1$. Обозначим через L максимум из L_1 и L'_1 и рассмотрим последовательность r_{L,l_2}. Первый ее член меньше, чем $\frac{p}{q}$. Так как $\lim_{l_2 \to \infty} r_{L,l_2} = \frac{p_2}{q_2} > \frac{p}{q}$, все члены, начиная с некоторого, больше, чем $\frac{p}{q}$. Учитывая, что шаг последовательности меньше $\frac{\gamma}{2}$, получаем, что в проколотой γ-окрестности $\frac{p}{q}$ есть значения $\frac{i(\varkappa)}{|\varkappa|}$.

Покажем, что если выполнено условие 2, то $\delta = \frac{p}{q}$, где q — длина цикла C, принадлежит спектру. Рассмотрим последовательность, состоящую из пути, ведущего в C (возможно, пустого), и k шагов по циклу C. Легко увидеть, что при стремлении k к бесконечности $\frac{i(\varkappa)}{|\varkappa|} \to \frac{p}{q}$. Так как δ не равно 0 и 1, для слов, длина которых является простым числом, значения $\frac{i(\varkappa)}{|\varkappa|}$ являются несократимыми дробями с знаменателями, равными рассматриваемым простым числам, получаем, что в любой проколотой окрестности δ имеются значения $\frac{i(\varkappa)}{|\varkappa|}$.

Покажем, что если выполнено условие 3, то 0 принадлежит спектру. Действительно, рассмотрим последовательность, состоящую из k обходов цикла C, все состояния которого безопасны, и пути, ведущего в небезопасное состояние. Так как одно из состояний небезопасно, $\frac{i(\varkappa)}{|\varkappa|} > 0$. Так как число небезопасных состояний конечно, при стремлении k к бесконечности $\frac{i(\varkappa)}{|\varkappa|}$ будет стремиться к 0.

Покажем, что если выполнено условие 4, то 1 принадлежит спектру. Действительно, рассмотрим последовательность, состоящую из k обходов цикла C, все состояния которого небезопасны, и пути, ведущего в безопасное состояние. Так как одно из состояний безопасно, и входит в состав $\varkappa$ (то есть это не первое вхождение начального состояния), $\frac{i(\varkappa)}{|\varkappa|} < 1$. Так как число безопасных состояний конечно, при стремлении k к бесконечности $\frac{i(\varkappa)}{|\varkappa|}$ будет стремиться к 1.

Покажем, что если для $\delta = \frac{p}{q}$ не выполнено ни одно из условий 1–4, то δ не принадлежит спектру. Если $\delta = 1$, то либо все циклы содержат

безопасные состояния, либо существуют циклы, состоящие из небезопасны состояний, но все пути, содержащие такие циклы, состоят только из небезопасных состояний. В первом случае значения $\frac{i(\varkappa)}{|\varkappa|}$ для всех слов, начиная с некоторой длины, отделены от 1, во втором случае значение 1 является изолированной точкой.

Если $\delta = 0$, то либо все циклы содержат небезопасные состояния, либо существуют циклы, состоящие из безопасных состояний, но все пути, содержащие такие циклы, состоят только из безопасных состояний. В первом случае значения $\frac{i(\varkappa)}{|\varkappa|}$ для всех слов, начиная с некоторой длины, отделены от 0, во втором случае значение 0 является изолированной точкой.

Рассмотрим случай $\delta \in (0; 1)$. Назовем компонентой диаграммы Мура подграф, порожденный множеством циклов. Два цикла попадают в одну компоненту тогда и только тогда, когда в диаграмме Мура есть путь, соединяющий эти циклы. Если $0 < \delta < 1$, то в каждой компоненте либо доля небезопасных состояний во всех циклах больше δ, либо либо доля небезопасных состояний во всех циклах меньше δ. Без ограничения общности рассмотрим первый случай. Пусть $\delta' > \delta$ — минимальная доля небезопасных состояний в циклах компоненты. Минимум существует и достигается на простом цикле, так как цикл, не являющийся простым, можно разделить на простые компоненты, причем минимальная доля небезопасных состояний в простой компоненте будет меньше или равна доли небезопасных состояний большого цикла. Рассмотрим множество частичных пределов $\frac{i(\varkappa)}{|\varkappa|}$ для данной компоненты. $\frac{\delta' - \delta}{2}$ будет нижней гранью этого множества, так как для всех слов, начиная с некоторой длины, $\frac{i(\varkappa)}{|\varkappa|} > \frac{\delta' - \delta}{2}$. Следовательно, в $\frac{\delta' - \delta}{2}$-окрестности δ имеется только конечное множество точек множества Δ. Рассматривая поочередно все компоненты, получаем искомое утверждение. □

5. Доказательство теоремы 3

Пусть $\varepsilon \in \mathbb{Q}$, $0 \leqslant \varepsilon < 1$, принадлежит спектру. Предположим, что существует конечный эксперимент, восстанавливающий параметры безопасности в этом случае. Пусть n — максимальная длина входного слова, поданного в процессе эксперимента. Выберем проколотую окрестность ε, в которой нет рациональных точек вида $\frac{p}{q}$, $p \leqslant n$, $q \leqslant n$. Выберем два элемента Δ, принадлежащих этой окрестности; обозначим их ε' и ε'', $\varepsilon' < \varepsilon''$. По лемме о склеивании значений ε, результаты эксперимента для ε' и ε'' совпадут. Легко увидеть, что ε'- и ε''-безопасные языки различаются. Дей-

ствительно, рассмотрим входное слово α, на котором достигается значение ε''. Слово α ε''-безопасно, но не ε'-безопасно. Следовательно, однозначное восстановление параметров невозможно — противоречие.

Пусть $\varepsilon = 1$, и не существует перехода из начального состояние в небезопасное состояние по слову длины 1. Предположим, что существует конечный эксперимент, восстанавливающий параметры безопасности в этом случае. Пусть n — максимальная длина входного слова, поданного в процессе эксперимента. Следовательно, значения $\dfrac{i(\varkappa)}{|\varkappa|}$ для поданных слов имеют вид $\dfrac{p}{q}$, где $p < q$ (так как при переходах на любом входном слове попадаем по крайней мере в одно безопасное состояние), $q \leqslant n$. По теореме 2, в диаграмме Мура есть цикл C, состоящий только из небезопасных состояний, связанный путем P с безопасным состоянием. Пусть такой путь состоит из l_1 безопасных и l_2 небезопасных состояний. Добавим к P k обходов цикла C и обозначим получившийся путь через P_k. Для P_k $\dfrac{i(\varkappa)}{|\varkappa|} \to 1$, $k \to \infty$, следовательно, можно выбрать слово, для которого $\dfrac{i(\varkappa)}{|\varkappa|} > \dfrac{n}{n+1}$. Такое слово 1-безопасно, но не $\dfrac{n}{n+1}$-безопасно, но эксперимент не различает 1- и $\dfrac{n}{n+1}$-безопасные языки — противоречие.

Рассмотрим следующий эксперимент. Если в диаграмме Мура есть переход из начального состояния в небезопасное по слову длины 1, сначала подадим на вход такое слово. Если оно окажется безопасным, $\varepsilon = 1$, и ε-безопасный язык представляет собой A^*.

Для каждой компоненты диаграммы Мура выделим циклы с минимальной и максимальной долей небезопасных состояний. Обозначим минимальную долю через λ_1, максимальную долю — через λ_2. Будем подавать на вход слова α_k и β_k, соответствующие k шагам обхода минимального и максимального цикла с учетом предпериодов и, возможно, постпериодов. Рассмотрим последовательность α_k; β_k рассматривается аналогично и подается параллельно с α_k. Постпериод добавляется, если существует такой постпериод, что доля небезопасных состояний в предпериоде и постпериоде больше доли небезопасных состояний в минимальном цикле, и выбирается минимальным по длине.

Пусть такой постпериод существует. Тогда возможны два случая. Если ε меньше доли небезопасных состояний в минимальном цикле, для некоторых k и $\gamma > 0$ $\dfrac{i(\varkappa)}{|\varkappa|}$ для α_k не превосходит доли небезопасных состояний минимального цикла минус γ, но α_k не является безопасным. Учитывая, что во всех словах, содержащихся в рассматриваемой компоненте, длина

которых не меньше некоторой константы, доля небезопасных состояний не меньше $\lambda_1 - \gamma$, получаем, что для данной компоненты достаточно проверить принадлежность ε-безопасному языку конечного множества слов, так как остальные слова заведомо не являются ε-безопасными. Предельная длина легко находится по конкретной компоненте диаграммы Мура. Во втором случае, учитывая теорему 2 и проводя аналогичные рассуждения для β_k, получаем, что для восстановления достаточно рассмотреть конечное множество слов, так как более длинные слова заведомо ε-безопасны.

Пусть такого постпериода не существует. Если $\lambda_1 = \lambda_2$, в этом случае рассматриваем последовательность β_k, пока $\dfrac{i(\varkappa)}{|\varkappa|} - \lambda_1 \geqslant \nu$, или пока слово не окажется безопасным. Если слово окажется безопасным, значит, все слова с длиной, большей некоторой константы, также безопасны. В противном случае все слова большей длины оказываются небезопасными (в силу условия 2 теоремы). Если $\lambda_1 < \lambda_2$, проводим аналогичные рассуждения с помощью последовательности α_k, выясняя, как соотносится ε с нижней границей, затем рассматриваем верхнюю границу и объединяем результаты. Легко увидеть, что при выполнении условий теоремы эксперимент окажется конечным. Теорема доказана. □

Литература

[1] *Галатенко А. В.* Автоматные модели защищенных компьютерных систем. Интеллектуальные системы, т. 11, вып. 1–4, Москва, 2007.

[2] *Кудрявцев В. Б.*, *Алешин С. В.*, *Подколзин А. С.* Введение в теорию автоматов. М.: Наука, 1985.

Анализ способов управления безопасностью информационных систем с помощью методов многокритериальной оптимизации и аппарата графов

П. Д. Зегжда, М. О. Калинин, Д. А. Москвин

Любая политика безопасности (ПБ), которую необходимо применить в информационной системе (в частности, в операционной системе Windows), как правило, представима в виде набора правил. Для реализации каждого правила ПБ администратор безопасности должен выполнить некоторое количество настроек, приняв при этом необходимое множество *управляющих решений*. Количество решений при этом не всегда соответствует количеству выполненных настроек. Например, для реализации правила ПБ вида

«Пользователь *User* должен иметь доступ на чтение к объектам каталога *C*:*Windows*»

администратору безопасности необходимо принять два решения: назначить право пользователю *User* на доступ к каталогу «*C*:*Windows*» и установить область действия прав «*Для этой папки, ее подпапок и файлов*». При этом операционная система автоматически устанавливает указанное право всем объектам каталога *C*:*Windows* [1]. Таким образом, для выполнения правила ПБ требуется два решения администратора безопасности и n выполненных настроек, где n — число объектов в каталоге *C*:*Windows*.

Изменение одной настройки безопасности (НБ) назовем *операцией*. Примерами операций являются: установка прав доступа пользователя к файлу, назначение пользователю привилегии и другие. Поскольку при настройке безопасности автоматизированным способом (например, скрипты) у администратора не возникает необходимости принимать решения по каждой настройке, то более важным становится количество операций, которое необходимо выполнить для реализации каждого правила ПБ, а также время выполнения настройки безопасности операционной системы. Пусть применение одного правила ПБ требует принятия администратором безопасности x решений или выполнения y операций, тогда $y = x \cdot m$, где

m — количество операций, которое необходимо выполнить для реализации принимаемых решений. Таким образом, возникает задача минимизации значения y при одновременном соблюдении следующих условий:

- ПБ не нарушается;
- система настраивается быстро и гибко.

Такой подход позволит обеспечить удовлетворительную скорость настройки безопасности и повысить эффективность управления безопасностью системы.

Для решения данной задачи авторами предложен метод повышения эффективности администрирования процессов управления безопасностью, основанный на количественной оценке сложности настройки и управления безопасностью системы, на применении многокритериальной оптимизации при настройке безопасности операционной системы с помощью графовых моделей.

Сложность настройки параметров безопасности системы, с точки зрения принятия решений администратором безопасности, соответствует *количеству принимаемых решений* (*КПР*), необходимых для выполнения всех правил ПБ: $= \sum_i x_i$, где x_i — количество решений для выполнения i-го правила ПБ. Чем меньше значение КПР, тем быстрее администратор может выполнить настройку параметров безопасности (далее именуется настройкой безопасности или НБ), и тем меньше вероятность возникновения ошибок администрирования.

Сложность управления безопасностью системы (СУБ) соответствует количеству операций, необходимых для модификации НБ в соответствии с правилами ПБ: $= \sum_i y_i z_i$, где y_i — количество операций для выполнения i-го правила ПБ; z_i — количество операций для модификации сторонних НБ. Сторонние НБ напрямую не связаны с реализацией правил ПБ, однако их необходимо настраивать, имея ввиду их связи с изменяемыми НБ. Например, доступ на чтение к файлу *file*1 предоставлен пользователю *User*1 с помощью привилегии «*Архивирование файлов и каталогов*». После изменения ПБ пользователь *User*1 не должен иметь доступа к файлу *file*1. Следовательно, у данного пользователя необходимо забрать имеющуюся привилегию. Однако, привилегия действует на все файлы, и пользователь потеряет доступ ко всем файлам. Тогда, в дополнение к отключению привилегии необходимо изменить другие НБ так, чтобы пользователь *User*1 не потерял доступ к файлам. Для этого, например, на все файлы, кроме *file*1, явно устанавливают право чтения. В данном случае, НБ всех файлов, кроме *file*1, являются сторонними.

Количество операций модификации сторонних НБ определяется формулой $z_i = \sum_j m_j$, где m_j — количество операций для выполнения j-ой настройки (для одной НБ). При этом количество операций модификации НБ равно количеству операций, необходимых для реализации решения по модификации этой НБ.

Таким образом, чем меньше сторонних НБ затрагивается в процессе управления безопасностью, тем меньше будет значение *СУБ*.

Относительное время (*ОВ*) настройки безопасности системы определяется настоящим (абсолютным) временем, которое требуется операционной системе для выполнения операций по установке НБ. Принимая во внимание тот факт, что время настройки напрямую зависит от производительности системы, используется относительное время, за единицу которого принимается время установки одной записи контроля доступа в списке контроля доступа объекта. Поскольку соотношение времени выполнения различных операций также зависит от аппаратного обеспечения системы, то при вычислении ОВ применяются усредненные значения.

Относительное время вычисляется по формуле $= \sum_i \beta_i y_i$, где y_i — количество операций для выполнения i-ого правила ПБ; β_i — весовой коэффициент, характеризующий время выполнения настройки и определяемый, как сумма весовых коэффициентов времени выполнения каждой операции для выполнения i-ого правила ПБ.

Оптимальной настройкой безопасности системы будем считать такую настройку, при которой одновременно оказываются минимальными величины КПР, СУБ и ОВ. *Критерием оптимизации* настройки безопасности системы будем называть условие минимизации одной из перечисленных величин. Тогда условие минимизации количества принимаемых решений назовем *КПР-критерием*, сложности модификации — *СУБ-критерием*, а относительное время настройки — *ОВ-критерием*.

Проблематика многокритериальной оптимизации относится к области многокритериальных задач принятия решений [2]. Многокритериальные задачи оптимизации вместе с множеством возможных (допустимых) решений $X \subset R^n$ включает набор целевых функций $f_1, f_2, \ldots, f_m$, при $m > 1$, заданных на множестве X. Каждая функция f_i описывает оптимизацию по одному из параметров. Многокритериальные задачи принятия решений условно можно разделить на две группы: многокритериальное математическое программирование и анализ решений по набору показателей. Анализ решений по набору показателей применяется в ситуациях с небольшим числом альтернатив в условиях неопределенности. Многокритериальное математическое программирование применимо при решении детерминиро-

ванных задач с большим числом возможных альтернатив. По этой причине оно хорошо подходит для выбора оптимального способа установки НБ в операционной системе.

Задача многокритериального математического программирования в общем виде представляется следующим образом:

$$\min\{f_1(x) = F_1\}, \quad \min\{f_2(x) = F_2\}, \quad \ldots, \quad \min\{f_k(x) = F_k\},$$

при $x \in X$, где X — множество допустимых значений переменных x; k — число целевых функций (критериев); F_i — значение i-го критерия (целевой функции). По существу, многокритериальная задача отличается от обычной задачи оптимизации наличием нескольких целевых функций вместо одной. В задачах многокритериального выбора решения выделяют область компромиссов (или решений, оптимальных по Парето) [3].

Существует несколько методов многокритериальной оптимизации [3]: принцип справедливого компромисса; принцип слабой оптимальности по Парето; принцип приближения по всем локальным критериям к идеальному решению; метод квазиоптимизации локальных критериев (метод последовательных уступок); метод свертывания векторного критерия в суперкритерий.

Для минимизации величин КПР, СУБ и ОВ применительно к НБ наиболее подходящим является метод свертывания векторного критерия в суперкритерий. Причина в том, что он позволяет изменять важность каждого критерия в зависимости от различных характеристик системы. Таким образом, будем вычислять *абсолютную сложность* настройки безопасности системы, которая учитывает КПР, СМ и ОВ по формуле $S_{\text{abs}} = \sum_{n=1}^{3} \alpha_n K_n$, где α_n — весовой коэффициент n-го критерия (n-й сложности); $K_1 = \sum_i x_i$ — КПР; $K_2 = \sum_i y_i z_i$ — СУБ; $K_3 = \sum_i \beta_i y_i$ — ОВ. Абсолютная сложность отражает количество решений администратора безопасности, выполняемых операций и время настройки безопасности системы. Следовательно, она напрямую зависит от объема настраиваемой системы, то есть от количества рабочих станций и серверов, субъектов и объектов доступа, от использования службы каталога Active Directory и установленных приложений. В этой связи, если сравнивать некачественно настроенную систему малого объема и настроенную систему большого объема, то всегда сложность для системы малого объема будет меньше. По этой причине абсолютную сложность следует применять только при сравнении схожих систем [4]. В том случае, если требуется сравнивать разные системы, то в этом случае необходимо использовать *относительную сложность*, которая учитывает количество НБ в системе

$S_{\text{rel}} = \dfrac{S_{\text{abs}}}{U+O} = \sum\limits_{n=1}^{3} \alpha_n K_n/(U+O)$, где U — количество настроенных пользователей и групп пользователей; O — количество настроенных объектов (учитываются не только защищаемые объекты операционной системы, но и объекты групповых политик, настройки приложений и другие характеристики). Далее под термином «сложность» понимается относительная сложность настройки безопасности системы, $S = S_{\text{rel}}$. Для оценки сложности по некоторой шкале применяется нормированная сложность $S_{\text{norm}} = 1 - \dfrac{1}{S}$.

Существуют различные способы выбора весовых коэффициентов α_i для каждого критерия оптимизации. Одним из них является назначение α_i в зависимости от относительной важности критериев. Чем «важнее» критерий, тем в большей степени он должен влиять на общую сложность настройки безопасности [5]. Каждый из трех критериев, используемых при расчете сложности, позволяет оптимизировать различные характеристики системы. По этой причине весовые коэффициенты каждого из критериев имеют разный смысл (табл. 2), который позволяют в зависимости от назначения и производительности системы и квалификации администратора назначать важность критериев. Смысловая нагрузка весовых коэффициентов: α_1 — КПР — квалификация администратора безопасности; α_2 — СМ — назначение и условия эксплуатации системы; α_3 — ОВ — производительность системы;

Будем считать, что чем меньше относительная сложность настройки системы по всем трем критериям с учетом их весовых коэффициентов, тем выше *степень оптимизации* настройки безопасности системы: $OPT = \dfrac{1}{S}$, где S — сложность. Допустимо ранжировать системы по степени оптимизации. Например: степень оптимизации меньше 0,05 — неудовлетворительно; от 0,05 до 0,15 — удовлетворительно; от 0,15 до 0,4 — хорошо; больше 0,4 — отлично.

Степень оптимизации никогда не может превышать значения 0,5, так как сложность модификации n настроек не может требовать меньше, чем n операций. В хаотично настроенных системах степень оптимизации ничтожно мала. Для нахождения минимального значения относительной сложности настройки безопасности системы применим аппарат теории графов и метод свертывания векторного критерия в суперкритерий. Для этого построим три графа, каждый из которых является графическим отображением одного из критериев оптимизации. Узлами графов будут устанавливаемые (модифицируемые) НБ. Тогда, задача минимизации сложности настройки безопасности сводится к нахождению минимального маршрута на каждом графе. После нахождения минимального маршрута по каждо-

му графу вычисляется длина этого маршрута, соответствующая сложности настройки системы. Далее выполняется суперпозиция графов и вычисляется оптимальный маршрут.

Таким образом, авторами предложен метод оценки оптимальной настройки и управления безопасностью информационных систем на основе аппарата графов с учетом заданных весовых коэффициентов критериев оптимизации.

Использование метода графов при решении задачи многокритериальной оптимизации позволяет в интуитивно понятной, визуальной, математически обоснованной графической форме выполнять поиск оптимального решения, а также доказывать его оптимальность. Формальная основа предложенного метода позволяет его запрограммировать и применить в системе автоматизированной настройки и управления безопасностью информационных систем. Разработка такого рода системы направлена на достижение быстрой и качественной автоматической настройки безопасности, что исключает возникновение конфигурационных нарушений безопасности, связанных с человеческим фактором.

Литература

[1] *Руссинович М.*, *Соломон Д.* Внутреннее устройство Microsoft Windows Server 2003, Windows XP и Windows 2000. Мастер-класс, СПб: Питер, 2005.

[2] *Штойер Р.* Многокритериальная оптимизация: теория, вычисления и приложения. М.: Радио и связь, 1992.

[3] *Ларичев О. И.* Теория и методы принятия решений. М.: Логос, 2000.

[4] *Лотов А. В.*, *Бушенков В. А.*, *Каменев В. А.*, *Черных О. Л.* Компьютер и поиск компромисса. Метод достижимых целей. М.: Наука, 1997.

[5] *Захаров И. Г.* Обоснование выбора. Теория практики. СПб: Судостроение, 2006.

Применение базовой ролевой ДП-модели для анализа условий передачи прав доступа

П. Н. Девянин

Для анализа безопасности управления доступом и информационными потоками между сущностями компьютерной системы (КС) с ролевым управлением доступом (РУД) построена базовая ролевая ДП-модель (далее, сокращенно, БР ДП-модель), основанная на семействе ролевых моделей *RBAC* [5, 3, 1] и семействе дискреционных и мандатных ДП-моделей [2]. БР ДП-модель позволяет анализировать условия передачи прав доступа и реализации информационных потоков по памяти и по времени с учетом доверенных и недоверенных субъект-сессий, функциональной корректности субъект-сессий и корректности субъект-сессий относительно сущностей, с учетом фактических ролей, прав доступа и возможных действий субъект-сессий.

В рамках БР ДП-модели задаются множества: объектов (O), контейнеров (C), сущностей (E), пользователей (U), доверенных пользователей (L_U), недоверенных пользователей (N_U), субъект-сессий пользователей (S), доверенных субъект-сессий (L_S), недоверенных субъект-сессий (N_S), ролей (R), административных ролей (AR), видов прав доступа ($R_r = \{read_r, write_r, append_r, execute_r, own_r\}$), видов доступа ($R_a = \{read_a, write_a, append_a, own_a\}$), видов информационных потоков ($R_f = \{write_m, write_t\}$), прав доступа к сущностям ($P \subset E \times R_r$), доступов субъект-сессий к сущностям ($A \subseteq S \times E \times R_a$), информационных потоков ($F \subseteq E \times E \times R_f$).

По аналогии с ролевыми моделями *RBAC* и базовой ДП-моделью задаются иерархии сущностей ($H_E\colon E \to 2^E$), ролей ($H_R\colon R \to 2^R$), административных ролей ($H_{AR}\colon AR \to 2^{AR}$), авторизованных ролей пользователей ($UA\colon U \to 2^R$), авторизованных административных ролей пользователей ($AUA\colon U \to 2^{AR}$), прав доступа ролей ($PA\colon R \to 2^P$), принадлежности субъект-сессии пользователю ($user\colon S \to U$), текущих ролей субъект-сессий ($roles\colon S \to 2^R \cup 2^{AR}$), администрирования прав доступа ролей ($can\_manage\_rights\colon AR \to 2^R$). Также задаются $G = (UA, AUA, PA, user, roles, A, F, H_R, H_{AR}, H_E, L_U, L_S)$ — состояние системы, $\Sigma(G^*, OP)$ — система (при этом G^* — множество всех возможных состояний, OP — мно-

жество правил преобразования состояний, $G \vdash_{op} G'$ — переход системы из состояния G в состояние G' с использованием правила преобразования состояний $op \in OP$), $\Sigma(G^*, OP, G_0)$ — система $\Sigma(G^*, OP)$ с начальным состоянием G_0.

Основные модели РУД, как правило, не содержат описания правил перехода КС из состояния в состояние. В то же время отсутствие четких правил перехода КС с РУД из состояния в состояние может также как в моделях КС с мандатным управлением доступом, построенным на основе модели Белла — ЛаПадулы [4], привести к разработке и использованию для анализа безопасности КС неадекватных ей формальных моделей. В рамках БР ДП-модели по аналогии с ДП-моделями КС с дискреционным или мандатным управлением доступом все правила преобразования состояний системы определены формально.

БР ДП-модель предназначена для анализа условий реализации в КС с РУД информационных потоков и в рамках нее не предполагается исследовать вопросы администрирования множества ролей, иерархии ролей, иерархии административных ролей, множеств авторизованных ролей пользователей, параметров ограничений. Таким образом, используется следующее предположение.

Предположение 1. В рамках БР ДП-модели на траекториях функционирования системы не изменяются значения множеств U, L_U, R и функции UA, AUA, H_R, H_{AR}. В БР ДП-модели не используются статические или динамические ограничения.

В БР ДП-модели пользователи или субъект-сессии могут быть доверенными или недоверенными. При этом в отличие от доверенных субъектов систем с дискреционным управлением доступом доверенные пользователи или субъект-сессии могут не обладать ролями, включающими все права доступа ко всем сущностям системы. Кроме того, в современных КС возможна реализация механизмов, позволяющих недоверенным субъектам выполнять функции преобразования данных с использованием доверенных субъектов. Например, в СУБД существует механизм триггеров, которые могут активизироваться в результате выполнения недоверенными субъектами операций над данными и функционировать от имени доверенных субъектов СУБД. При этом некоторые доверенные субъекты могут участвовать в реализации информационных потоков по времени, а недоверенные субъекты не получают никаких прав доступа к активизированным ими доверенным субъектам и могут только использовать полученные доверенными субъектами результаты операций над данными.

Чтобы не усложнять описание правил преобразования состояний системы, в результате выполнения которых создаются субъект-сессии, целесообразно считать, что недоверенные пользователи или субъект-сессии не

создают доверенных субъект-сессий, доверенные пользователи или субъект-сессии не создают недоверенных субъект-сессий, а все используемые для преобразования данных недоверенными субъект-сессиями доверенные субъект-сессии уже существуют во всех состояний системы. Кроме того, будем считать, что в системе не рассматриваются недоверенные пользователи или недоверенные субъект-сессии, которые не могут создать субъект-сессию. Таким образом, используются следующие предположение и определение.

Предположение 2. Каждый пользователь или субъект-сессия системы $\Sigma(G^*, OP)$ вне зависимости от имеющихся у них авторизованных ролей являются либо доверенными, либо недоверенными. Доверенные пользователи или субъект-сессии не создают новых субъект-сессий. Каждый недоверенный пользователь или субъект-сессия могут создать только недоверенную субъект-сессию.

Определение 1. Доверенную субъект-сессию назовем корректной относительно информационных потоков по времени, если она не участвует в их реализации.

Используем обозначения: $LF_S \subset L_S$ — множество доверенных субъект-сессий корректных относительно информационных потоков по времени, $NF_S \subset L_S$ — множество доверенных субъект-сессий некорректных относительно информационных потоков по времени. В рамках предположений 1 и 2 будем использовать следующее сокращенное обозначение для состояния системы $G = (PA, user, roles, A, F, H_E)$.

Используем предположения аналогичные сделанным в рамках ФАС ДП-модели.

Предположение 3. Только информационный поток по памяти к сущности, функционально ассоциированной с субъект-сессией, приводит к изменению вида преобразования данных, реализуемого этим субъект-сессией. Множество сущностей, функционально ассоциированных с субъект-сессией, не изменяется в процессе функционирования системы.

Предположение 4. При создании субъект-сессии s множество функционально ассоциированных с ней сущностей задается только в зависимости от сущности, из которой создается субъект-сессия s, и пользователя, который либо создает субъект-сессию s, либо от имени которого другая субъект-сессия создает субъект-сессию s.

Используем обозначения: $[s] \subset E \cup U$ — множество сущностей, функционально ассоциированных с субъект-сессией s (при этом по определению выполняется условие $s \in [s]$), и пользователей, каждый из которых может создать субъект-сессию, являющуюся функционально ассоциированной сущностью с субъект-сессией s; $fa\colon U \times E \to 2^E \cup 2^U$ — функ-

ция, задающая множества сущностей, функционально ассоциированных с субъект-сессией.

Определение 2. Доверенную субъект-сессию y назовем функционально корректной, если во множество функционально ассоциированных с ней сущностей $[y]$ не входят недоверенные субъект-сессии.

Определение 3. Доверенную субъект-сессию y назовем корректной относительно доверенной субъект-сессии y' и сущности e, если субъект-сессия y не реализует информационный поток по памяти от сущности e к сущности e', функционально ассоциированной с доверенной субъект-сессией y'.

Используем обозначение: $y(E) \subset E \times L_S$ — множество пар вида доверенная субъект-сессия и сущность, относительно которых корректна доверенная субъект-сессия y.

В современных КС при взаимодействии субъектов обмен данными между ними происходит, как правило, через сущности, не являющиеся субъектами, при этом субъекты не могут обладать правами доступа друг к другу за исключением права доступа владения.

В КС с РУД право доступа к сущности может быть получено субъект-сессией только через обладание ролью, содержащей данное право. Реализация субъект-сессией s_1 информационного потока по памяти на сущность, функционально ассоциированную с субъект-сессией s_2, позволит субъект-сессии s_1 получить контроль над субъект-сессией s_2, включая возможность использовать права доступа ролей, которыми обладает s_2, и возможность использовать информационные потоки, в реализации которых участвует субъект-сессия s_2. При этом множество текущих ролей субъект-сессии s_1, как правило, останется неизменным. Кроме того, в современных КС часто для изменения субъект-сессией множества своих текущих ролей требуется ввод аутентификационных данных пользователем, от имени которого функционирует субъект-сессия. Следовательно, контроль субъект-сессии s_1 над субъект-сессией s_2, не позволяет субъект-сессии s_1 изменять множество текущих ролей субъект-сессии s_2. Также следует отметить, что, получив контроль над субъект-сессией s_2, субъ-ект-сессия s_1 может, используя административные роли субъект-сессии s_2, осуществлять только те действия над ролями и сущностями (передача роли прав доступа к сущности, создание новой сущности внутри сущности контейнера, активизация новой субъект-сессии из сущности), которые позволяют ему выполнять права доступа ролей субъект-сессии s_2. Например, если в системе существуют различные субъект-сессия s_2, обладающая административной ролью для передачи прав доступа роли r, и субъект-сессия s_3 с ролью, имеющей право доступа (y, own_r), то субъект-сессия s_1, получившая контроль над субъект-сессиями s_2 и s_3, не сможет передать роли

r права доступа к сущности y. Таким образом, используется следующее предположение.

Предположение 5. Субъект-сессии могут иметь друг другу только доступ владения own_a. Роли могут обладать к субъект-сессиям только правом доступа владения own_r. Если субъект-сессия s_1 реализовала информационный поток по памяти от себя к сущности, функционально ассоциированной с другой субъект-сессией s_2, то субъект-сессия s_1 получает:

- доступ владения own_a к субъект-сессии s_2;
- возможность использовать роли из множества ролей $roles(s_2)$ (при этом субъект-сессия s_1 не может изменить множество текущих ролей $roles(s_2)$);
- возможность получать доступ владения own_a к субъект-сессиям, доступом владения к которым обладает субъект-сессия s_2;
- возможность использовать административные роли субъект-сессии s_2 для осуществления действий над ролями и сущностями, которые позволяют ей изменять права доступа ролей субъект-сессии s_2;
- возможность использовать информационные потоки, в реализации которых участвует субъект-сессия s_2.

Используем обозначения:

- $de\_facto\_roles\colon S \to 2^{R \cup AR}$ — функция фактических текущих ролей субъект-сессий, при этом по определению в каждом состоянии системы $G = (PA, user, roles, A, F, H_E)$ для каждой субъект-сессии $s_1 \in S$ выполняется равенство: $de\_facto\_roles(s_1) = roles(s_1) \cup \{r \in \in R \cup AR\colon$ существует $s_2 \in S$, $(s_1, s_2, own_a) \in A$ и $r \in roles(s_2)\}$;
- $de\_facto\_rights\colon S \to 2^P$ — функция фактических текущих прав доступа субъект-сессий, при этом по определению в каждом состоянии системы $G = (PA, user, roles, A, F, H_E)$ для каждой субъект-сессии $s \in S$ выполняется равенство: $de\_facto\_rights(s) = \{p \in \in P\colon$ существует $r \in de\_facto\_roles(s)$ и $p \in PA(r)\}$;
- $de\_facto\_actions\colon S \to 2^P \times 2^R$ — функция фактических возможных действий субъект-сессий, при этом по определению в каждом состоянии системы $G = (PA, user, roles, A, F, H_E)$ для каждой субъект-сессии $s_1 \in S$ выполняется равенство: $de\_facto\_actions(s_1) = = (PA(roles(s_1)) \times can\_manage\_rights(roles(s_1) \cap AR)) \cup \{(p,r) \in P \times R\colon$ существует $s_2 \in S, (s_1, s_2, own_a) \in A, r \in can\_manage\_rights(roles(s_2) \cap \cap AR)$ и $p \in PA(roles(s_2))\}$.

В рамках БР ДП-модели используются следующие 20 правил преобразования состояний:

$take\_role(x, r)$, $remove\_role(x, r)$, $grant\_right(x, r, (y, \alpha_r))$, $remove\_right(x, r, (y, \alpha_r))$, $create\_entity(x, r, y, z)$, $create\_first\_session(u, r, y, z)$, $create\_session(x, r, y, z)$, $delete\_entity(x, y, z)$, $rename\_entity(x, y, z)$, $control(x, y, z)$, $access\_own(x, y)$, $take\_access\_own(x, y, z)$, $access\_read(x, y)$, $access\_write(x, y)$, $access\_append(x, y)$, $flow(x, y, y', z)$, $find(x, y, z)$, $post(x, y, z)$, $pass(x, y, z)$, $take\_flow(x, y)$.

Например, условия применения правила $control(x, y, z)$ БР ДП-модели аналогичны условиям применения правила $control(x, y, z)$ ФАС ДП-модели и может быть применено субъект-сессией x для получения доступа владения own_a к субъект-сессии y. При этом субъект-сессия x должна либо входить во множество сущностей, функционально ассоциированных с субъект-сессией y, либо реализовать информационный поток по памяти к сущности z, функционально ассоциированной с субъект-сессией y, либо обладать доступом владения к субъект-сессии z, функционально ассоциированной с субъект-сессией y (рис. 1).

Правило $take\_flow(x, y)$ позволяет субъект-сессии x, получившей доступ владения к субъект-сессии y, использовать все информационные потоки по памяти или по времени, в реализации которых участвует субъект-сессия y (рис. 2).

Зависимость условий и результатов применения правил преобразования состояний БР ДП-модели показана на рис. 3.

Определение 4. Назовем правило преобразования состояний монотонным, если его применение не приводит к удалению из состояний: ролей из множества текущих ролей субъект-сессии, прав доступа ролей к сущностям, субъект-сессий или сущностей, доступов субъект-сессий к сущностям, информационных потоков.

Монотонными являются правила:

$take\_role(x, r)$, $grant\_right(x, r, (y, \alpha_r))$, $create\_entity(x, r, y, z)$, $create\_first\_session(u, r, y, z)$, $create\_session(x, r, y, z)$, $rename\_entity(x, y, z)$, $control(x, y, z)$, $access\_own(x, y)$, $take\_access\_own(x, y, z)$, $access\_read(x, y)$, $access\_write(x, y)$, $access\_append(x, y)$, $flow(x, y, y', z)$, $find(x, y, z)$, $post(x, y, z)$, $pass(x, y, z)$, $take\_flow(x, y)$.

Немонотонными являются правила:

$remove\_role(x, r)$, $remove\_right(x, r, (y, \alpha_r))$, $delete\_entity(x, y, z)$.

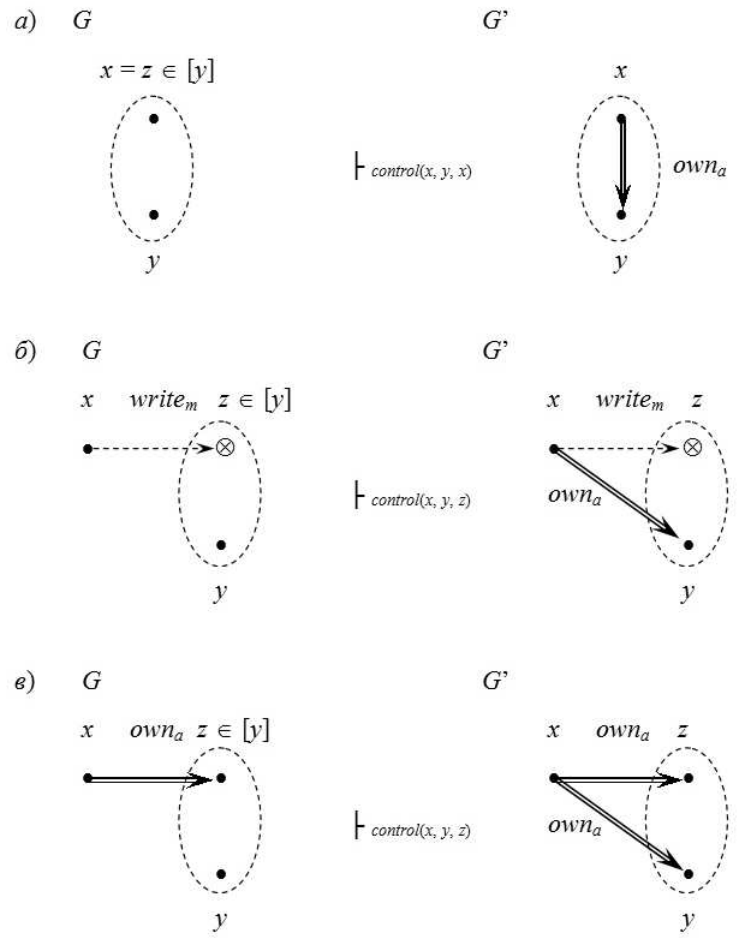


Рис. 1. Применение правила $control(x, y, z)$ в следующих случаев: а) $x = z \in [y]$; б) $z \in [y]$ и $(x, z, write_m) \in F$; в) $z \in [y]$ и $(x, z, own_a) \in A$

В рамках БР ДП-модели доказано утверждение, что при анализе условий передачи прав доступа, реализации информационных потоков по памяти или по времени достаточно рассматривать только монотонные правила преобразования состояний.

Применим БР ДП-модель для анализа случая, когда для передачи права доступа ролей непосредственно взаимодействуют только две субъект-сессии.

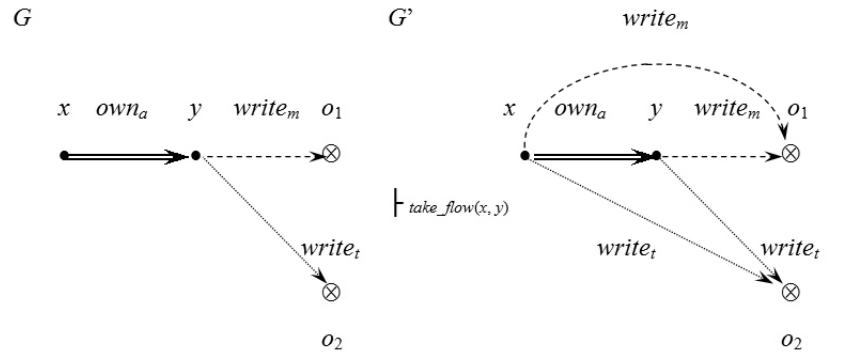


Рис. 2. Пример применения правила *take_flow*(*x*, *y*)

В рамках БР ДП-модели целесообразно использовать подход, примененный в ДП-моделях КС с дискреционным или мандатным управлением доступом для анализа систем, в которых доверенные и недоверенные субъекты не кооперируют между собой при передаче прав доступа или реализации информационных потоков. При этом важно учесть следующие существенные особенности БР ДП-модели.

В ДП-моделях КС с дискреционным управлением доступом предполагалось, что доверенные субъекты имеют все права доступа ко всем сущностям системы. Следовательно, при анализе безопасности системы достаточно определить условия, при выполнении которых один из недоверенных субъектов может получить право доступа владения к доверенному субъекту, после чего недоверенный субъект получает все права доступа ко всем сущностям КС, и дальнейший анализ условий получения другими недоверенными субъектами прав доступа к сущностям КС не имеет смысла. В рамках БР ДП-модели в произвольная доверенная субъект-сессия может не иметь ролей, обладающих в совокупности всеми правами доступа ко всем сущностям. Таким образом, при анализе условий получения недоверенной субъект-сессией фактической роли, обладающей заданным правом доступа к сущности, может оказаться недостаточным только исследование случая, когда недоверенная субъект-сессия реализовала доступ владения к некоторой доверенной субъект-сессии или получила фактические права доступа к некоторой сущности.

В существующих КС в случае, когда недоверенная субъект получает доступ владения к доверенному субъекту, последний может осуществлять в системе действия, запрещенные для доверенных субъектов в рамках ДП-моделей КС с дискреционным управлением доступом на траекториях без кооперации доверенных и недоверенных субъектов для передачи прав

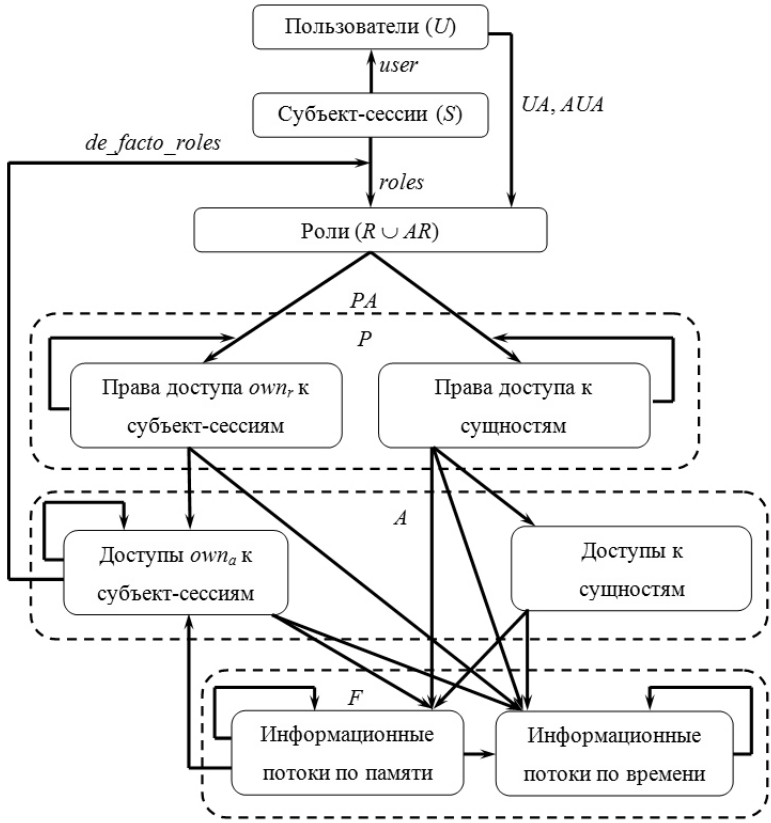


Рис. 3. Зависимость условий и результатов применения правил преобразования состояний БР ДП-модели

доступа и реализации информационных потоков (например, доверенный субъект может давать недоверенным субъектам права доступа к сущностям). В то же время условия применения правил преобразования состояний позволяют недоверенной субъект-сессии, получившей доступ владения к доверенной субъект-сессии, самостоятельно выполнять в системе все действия (кроме изменения множества текущих ролей) от ее имени. Таким образом, в рамках БР ДП-модели для анализа условий передачи прав доступа или реализации информационных потоков возможно исследование только траекторий без кооперации доверенных и недоверенных субъект-сессий.

В ДП-моделях КС с дискреционным или мандатным управлением доступом не анализировались пользователи системы. В рамках БР ДП-модели рассматриваются доверенные и недоверенные пользователи, от имени которых активизируются и функционируют субъект-сессии.

В ДП-моделях КС с дискреционным управлением доступом рассматривался случай, когда хотя бы один недоверенный субъект может получить право доступа владения к доверенному субъекту (в данном случае предикат $can\_share\_own(x, y, G_0, L_S)$ является истинным). В рамках БР ДП-модели возможен случай, когда функционирующая от имени заданного недоверенного пользователя субъект-сессия пытается получить фактическое право доступа к заданной сущности. При этом другие существующие в системе доверенные или недоверенные субъект-сессии могут иметь роли, обладающие данным правом доступа к сущности, но могут не иметь ролей, необходимых для его передачи. Следовательно, целесообразно учесть случай, когда для получения фактического права доступа к сущности недоверенная субъект-сессия получает доступы владения к доверенным и недоверенным субъект-сессиям.

Определение 5. Назовем траекторию функционирования системы $\Sigma(G^*, OP)$ траекторией без кооперации доверенных и недоверенных субъект-сессий для передачи прав доступа, если при ее реализации используются только монотонные правила преобразования состояний, и доверенные субъект-сессии не берут роли во множество текущих ролей, не дают другим ролям права доступа к сущностям, не получают доступ владения к субъект-сессиям.

Определение 6. Пусть G_0 — состояние системы $\Sigma(G^*, OP)$, в котором существуют пользователь $x \in U_0$ и право доступа к сущности $(e, \alpha) \in \in P_0$. Определим предикат $can\_share((e, \alpha), x, G_0)$, который будет истинным тогда и только тогда, когда существуют состояния $G_1, \ldots, G_N$ и правила преобразования состояний $op_1, \ldots, op_N$, где $N \geqslant 0$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ является траекторией без кооперации доверенных и недоверенных субъект-сессий для передачи прав доступа, и существует субъект-сессия $s_x \in S_N$ такая, что $user_N(s_x) = x$ и право доступа к сущности $(e, \alpha) \in de\_facto\_rights_N(s_x)$.

Для упрощения записи алгоритмически проверяемых необходимых и достаточных условий истинности предиката $can\_share((y, \alpha), u, G_0)$ используем следующие определения.

Определение 7. Пусть G — состояние системы $\Sigma(G^*, OP)$, в котором существуют субъект-сессии или недоверенные пользователи $x, y \in N_U \cup S$. Определим предикат $directly\_access\_own(x, y, G)$, который будет истинным тогда и только тогда, когда или $x = y$, или выполняется одно из условий.

Условие 1. Если $y \in N_U$ и $x \in N_U$, то существуют сущность $e_y \in E$ и роль $r_y \in R$ такие, что

$$(e_y, execute_r) \in PA(UA(y)), \quad r_y \in can\_manage\_rights(AUA(y)),$$

и выполняется одно из условий:

- $r_y \in UA(x)$;
- $x \in fa(y, e_y)$;
- существует сущность $e \in E$ такая, что $(e, \beta) \in PA(UA(x))$, где $\beta \in \{write_r, append_r, own_r\}$, и или $e \in fa(y, e_y)$, или $(e, \gamma) \in PA(UA(y))$, где $\gamma \in \{read_r, own_r\}$.

Условие 2. Если $y \in N_U$ и $x \in N_S \cap S$, то существуют сущность $e_y \in E$ и роль $r_y \in R$ такие, что

$$(e_y, execute_r) \in PA(UA(y)), \quad r_y \in can\_manage\_rights(AUA(y)),$$

и выполняется одно из условий:

- $r_y \in UA(user(x))$;
- $x \in fa(y, e_y)$;
- существует сущность $e \in E$ такая, что $(e, \beta) \in PA(UA(user(x))$, где $\beta \in \{write_r, append_r, own_r\}$, или $(x, e, write_m) \in F$, и или $e \in fa(y, e_y)$, или $(e, \gamma) \in PA(UA(y))$, где $\gamma \in \{read_r, own_r\}$.

Условие 3. Если $y \in N_U$ и $x \in L_S \cap S$, то существуют сущность $e_y \in E$ и роль $r_y \in R$ такие, что

$$(e_y, execute_r) \in PA(UA(y)), \quad r_y \in can\_manage\_rights(AUA(y)),$$

и выполняется одно из условий:

- $r_y \in roles(x)$;
- $x \in fa(y, e_y)$;
- существует сущность $e \in E$ такая, что $(e, \beta) \in PA(roles(x))$, где $\beta \in \{write_r, append_r\}$, или $(x, e, write_m) \in F$, и или $e \in fa(y, e_y)$, или $(e, \gamma) \in PA(UA(y))$, где $\gamma \in \{read_r, own_r\}$.

Условие 4. Если $y \in S$ и $x \in N_U$, то выполняется одно из условий:

- $(y, own_r) \in PA(UA(x))$;
- $x \in [y]$;

- существует сущность $e \in E$ такая, что $(e, \beta) \in PA(UA(x))$, где $\beta \in \in \{write_r, append_r, own_r\}$, и или $e \in [y]$, или $y \in N_S \cap S$, $(e, \gamma) \in \in PA(UA(user(y)))$, где $\gamma \in \{read_r, own_r\}$, или $y \in L_S \cap S$, $(e, read_r) \in \in PA(roles(y))$, $(y, e) \notin y(E)$.

Условие 5. Если $y \in S$ и $x \in N_S \cap S$, то выполняется одно из условий:

- $(y, own_r) \in PA(UA(user(x)))$;
- $x \in [y]$;
- $(x, y, own_a) \in A$;
- существует сущность $e \in E$ такая, что $(e, \beta) \in PA(UA(user(x))$, где $\beta \in \in \{write_r, append_r, own_r\}$, или $(x, e, write_m) \in F$, и или $e \in [y]$, или $y \in N_S \cap S$, $(e, \gamma) \in PA(UA(user(y)))$, где $\gamma \in \{read_r, own_r\}$, или $y \in \in L_S \cap S$, $(e, read_r) \in PA(roles(y))$, $(y, e) \notin y(E)$.

Условие 6. Если $y \in S$ и $x \in L_S \cap S$, то выполняется одно из условий:

- $(y, own_r) \in PA(roles(x))$;
- $x \in [y]$;
- $(x, y, own_a) \in A$;
- существует сущность $e \in E$ такая, что $(e, \beta) \in PA(roles(x))$, где $\beta \in \in \{write_r, append_r\}$, или $(x, e, write_m) \in F$, и выполняется одно из условий:
  - $y \in N_S \cap S$ и или $e \in [y]$, или $(e, \gamma) \in PA(UA(user(y)))$, где $\gamma \in \in \{read_r, own_r\}$;
  - $y \in L_S \cap S$ и или $e \in [y]$, $(x, x) \notin y(E)$, или $(e, read_r) \in PA(roles(y))$, $(y, e) \notin y(E)$.

Определение 8. Пусть G — состояние системы $\Sigma(G^*, OP)$, в котором недоверенная субъект-сессия или недоверенный пользователь $x \in N_U \cup \cup (N_S \cap S)$, субъект-сессия или недоверенный пользователь $y \in N_U \cup S$. Определим предикат $directly\_grant\_right(x, y, G)$, который будет истинным тогда и только тогда, когда или $x = y$, или выполняется одно из условий.

Условие 1. Если $y \in N_U$ и $x \in N_U$, то существует роль

$$r \in can\_manage\_rights(AUA(x)) \cap UA(y).$$

Условие 2. Если $y \in N_U$ и $x \in N_S \cap S$, то существуют роль

$$r \in can\_manage\_rights(AUA(user(x))) \cap UA(y).$$

Условие 3. Если $y \in S$ и $x \in N_U$, то выполняется одно из условий:

- $y \in N_S \cap S$ и существуют роль $r \in can\_manage\_rights(AUA(x)) \cap \cap UA(user(y))$;
- $y \in L_S \cap S$ и существуют роль $r \in can\_manage\_rights(AUA(x)) \cap \cap roles(y)$.

Условие 4. Если $y \in S$ и $x \in N_S \cap S$, то выполняется одно из условий:

- $y \in N_S \cap S$ и существуют роль

$$r \in can\_manage\_rights(AUA(user(x))) \cap UA(user(y));$$

- $y \in L_S \cap S$ и существуют роль

$$r \in can\_manage\_rights(AUA(user(x))) \cap roles(y).$$

Утверждение 1. *Пусть G_0 — состояние системы $\Sigma(G^*, OP)$, в котором существуют недоверенный пользователь или недоверенная субъект-сессия $x \in N_U \cup N_S \cap S_0)$, субъект-сессия или недоверенный пользователь $y \in N_U \cup S_0$ и право доступа к сущности $(e, \alpha) \in P_0$. Пусть истинен предикат $directly\_access\_own(x, y, G_0)$ и выполняется одно из условий*:

- *если $y \in N_U$, то $(e, \delta) \in PA_0(UA_0(y))$, где $\delta \in \{\alpha, own_r\}$*;
- *если $y \in N_S \cap S_0$, то $(e, \delta) \in PA_0(UA_0(user_0(y)))$, где $\delta \in \{\alpha, own_r\}$*;
- *если $y \in L_S \cap S_0$ и $x \in N_U$, то либо $(e, \alpha) \in PA_0(roles_0(y))$, либо $(e, own_r) \in PA_0(roles_0(y))$ и $can\_manage\_rights(roles_0(y) \cap AR) \cap \cap (roles_0(y) \cup UA_0(x)) \neq \varnothing$*;
- *если $y \in L_S \cap S_0$ и $x \in N_S \cap S_0$, то либо $(e, \alpha) \in PA_0(roles_0(y))$, либо $(e, own_r) \in PA_0(roles_0(y))$ и $can\_manage\_rights(roles_0(y) \cap AR) \cap \cap (roles_0(y) \cup UA_0(user_0(x))) \neq \varnothing$.*

Тогда выполняется одно из условий.

Условие 1. *Если $x \in N_U$, то истинен предикат $can\_share((e, \alpha), x, G_0)$.*

Условие 2. *Если $x \in N_S \cap S_0$, то истинен предикат $can\_share((e, \alpha), user_0(x), G_0)$.*

Утверждение 2. *Пусть G_0 — состояние системы $\Sigma(G^*, OP)$, в котором существуют недоверенный пользователь или недоверенная субъект-сессия $x \in N_U \cup (N_S \cap S_0)$, субъект-сессия или недоверенный пользователь $y \in N_U \cup S_0$ и право доступа к сущности $(e, \alpha) \in P_0$. Пусть истинен предикат $directly\_grant\_right(x, y, G_0)$ и выполняется одно из условий*:

- *если $x \in N_U$, то $(e, own_r) \in PA_0(UA_0(x))$*;

- *если* $x \in N_S \cap S_0$, *то* $(e, own_r) \in PA_0(UA_0(user_0(x)))$.

Тогда выполняется одно из условий.

Условие 1. *Если* $y \in N_U$, *то истинен предикат* $can\_share((e, \alpha), y, G_0)$.

Условие 2. *Если* $y \in S_0$, *то истинен предикат* $can\_share((e, \alpha), user_0(y), G_0)$.

Определим и обоснуем алгоритмически проверяемые необходимые и достаточные условия истинности предиката $can\_share((y, \alpha), u, G_0)$ для случая, когда в системе существует только два пользователя или две субъект-сессии.

Определение 9. Пусть G — состояние системы $\Sigma(G^*, OP)$, в котором существует пользователь $x \in N_U$ и право доступа к сущности $(e, \alpha) \in P$. Определим предикат $directly\_can\_share((e, \alpha), x, G)$, который будет истинным тогда и только тогда, когда существует пользователь $y \in U$, и выполняется одно из условий, где $\delta \in \{\alpha, own_r\}$.

Условие 1. Выполняется одно из условий:

- пользователь $y \in N_U$, право доступа $(e, \delta) \in PA(UA(y))$, и истинен предикат $directly\_access\_own(x, y, G)$;
- существует недоверенная субъект-сессия $s_y \in N_S \cap S$ такая, что $user(s_y) = y$, истинен предикат $directly\_access\_own(x, s_y, G)$, и $(e, \delta) \in PA(UA(user(s_y)))$.

Условие 2. Существует недоверенная субъект-сессия $s_x \in N_S \cap S$ такая, что $user(s_x) = x$, и выполняется одно из условий:

- пользователь $y \in N_U$, право доступа $(e, \delta) \in PA(UA(y))$, и истинен предикат $directly\_access\_own(s_x, y, G)$;
- существует недоверенная субъект-сессия $s_y \in N_S \cap S$ такая, что $user(s_y) = y$, истинен предикат $directly\_access\_own(s_x, s_y, G)$, и $(e, \delta) \in PA(UA(user(s_y)))$.

Условие 3. Существует доверенная субъект-сессия $s_y \in L_S \cap S$ такая, что $user(s_y) = y$, право доступа $(e, \alpha) \in PA(roles(s_y))$, и выполняется одно из условий:

- истинен предикат $directly\_access\_own(x, s_y, G)$;
- существует недоверенная субъект-сессия $s_x \in N_S \cap S$ такая, что $user(s_x) = x$, и истинен предикат $directly\_access\_own(s_x, s_y, G)$.

Условие 4. Если $\alpha \neq own_r$, то существует доверенная субъект-сессия $s_y \in L_S \cap S$ такая, что $user(s_y) = y$, право доступа $(e, own_r) \in PA(roles(s_y))$, $can\_manage\_rights(roles(s_y) \cap AR) \cap (roles(s_y) \cup UA(x)) \neq \varnothing$, и выполняется одно из условий:

- истинен предикат $directly\_access\_own(x, s_y, G)$;
- существует недоверенная субъект-сессия $s_x \in N_S \cap S$ такая, что $user(s_x) = x$, и истинен предикат $directly\_access\_own(s_x, s_y, G)$.

Условие 5. Пользователь $y \in N_U$, право доступа $(e, own_r) \in PA(UA(y))$, и выполняется одно из условий:

- пользователь $x \in N_U$, и истинен предикат $directly\_grant\_right(y, x, G)$;
- существует субъект-сессия $s_x \in N_S \cap S$ такая, что $user(s_x) = x$, и истинен предикат $directly\_grant\_right(y, s_x, G)$.

Условие 6. Существует субъект-сессия $s_y \in N_S \cap S$ такая, что $user(s_y) = y$, право доступа $(e, own_r) \in PA(UA(user(s_y)))$, и выполняется одно из условий:

- пользователь $x \in N_U$, и истинен предикат $directly\_grant\_right(s_y, x, G)$;
- существует субъект-сессия $s_x \in N_S \cap S$ такая, что $user(s_x) = x$, и истинен предикат $directly\_grant\_right(s_y, s_x, G)$.

Утверждение 3. *Пусть G_0 — состояние системы $\Sigma(G^*, OP)$, в котором существует недоверенный пользователь $x \in N_U$ и право доступа к сущности $(e, \alpha) \in P_0$, и истинен $directly\_can\_share((e, \alpha), x, G_0)$. Тогда истинен предикат $can\_share((e, \alpha), x, G_0)$.*

Теорема 1. *Пусть G_0 — состояние системы $\Sigma(G^*, OP)$, в котором выполняются условия $|U_0| \leqslant 2$, $x \in N_U$, для каждого пользователя $u \in U_0$ верно неравенство $|user_0^{-1}(u)| \leqslant 1$, и существует право доступа к сущности $(e, \alpha) \in P_0$. Предикат $can\_share((e, \alpha), x, G_0)$ является истинным тогда и только тогда, когда является истинным предикат $directly\_can\_share((e, \alpha), x, G_0)$.*

Доказательство. Докажем достаточность выполнения условий теоремы для истинности предиката $can\_share((e, \alpha), x, G_0)$. Из доказательства утверждений 1–3 следует, что при истинности предиката $directly\_can\_share((e, \alpha), x, G_0)$ для передачи прав доступа достаточно наличия в системе не более двух пользователей, каждый из которых либо создает, либо имеет не более одной субъект-сессии. Значит, утверждения 1–3 справедливы при выполнении условий теоремы: $|U_0| \leqslant 2$, и для каждого пользователя $u \in U_0$ верно неравенство $|user_0^{-1}(u)| \leqslant 1$. Таким образом, по утверждению 3 предикат $can\_share((e, \alpha), x, G_0)$ является истинным.

Докажем необходимость выполнения условий теоремы для истинности предиката $can\_share((e, \alpha), x, G_0)$. По определению 6 существуют состояния $G_1, \ldots, G_N$ и правила преобразования состояний $op_1, \ldots, op_N$, где

$N \geqslant 0$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ является траекторией без кооперации доверенных и недоверенных субъект-сессий для передачи прав доступа, и существует субъект-сессия $s_x \in S_N$ такая, что $user_N(s_x) = x$ и право доступа к сущности $(e, \alpha) \in de\_facto\_rights_N(s_x)$.

Среди всех траекторий выберем ту, у которой длина N является минимальной. Проведем доказательство индукцией по длине траекторий N.

Пусть $N = 0$, тогда $(e, \alpha) \in de\_facto\_rights_0(s_x)$. По определению функции $de\_facto\_rights_0(s_x)$ возможны два случая.

Первый случай: право доступа

$$(e, \alpha) \in PA_0(roles_0(s_x)) \subset PA_0(UA_0(user_0(s_x))).$$

Положим $y = x$, $s_y = s_x$. Тогда предикат $directly\_access\_own(s_x, s_x, G_0)$ истинен, выполнено условие 2 определения 9 и является истинным предикат $directly\_can\_share((e, \alpha), x, G_0)$.

Второй случай: существует субъект-сессия $s_y \in S_0$ такая, что $(s_x, s_y, own_a) \in A_0$, $r_e \in roles_0(s_y)$, и

$$(e, \alpha) \in PA_0(r_e) \subset PA_0(roles_0(s_y)) \subset PA_0(UA_0(user_0(s_y))).$$

Тогда выполнено условие 5 определения 7, и является истинным предикат $directly\_access\_own(s_x, s_y, G_0)$. Положим $y = user_0(s_y)$, следовательно, выполнено условие 2 определения 9 и является истинным предикат $directly\_can\_share((e, \alpha), x, G_0)$.

Пусть $N = 1$, тогда из минимальности N следует, что выполняется условие $(e, \alpha) \notin de\_facto\_rights_0(s_x)$. По определению функции $de\_facto\_rights_1(s_x)$ возможны два случая.

Первый случай: право доступа $(e, \alpha) \in PA_1(roles_1(s_x))$. Тогда $(e, \alpha) \notin \notin PA_0(roles_0(s_x))$ и существует роль $r_e \in roles_1(s_x)$ такая, что $(e, \alpha) \in \in PA_1(r_e)$, и из определения правил преобразования состояний следует, что выполняется одно из трех условий.

Первое условие: $op_1 = take\_role(s_x, r_e)$. Тогда $r_e \in UA_0(user_0(s_x)) = = UA_0(x)$, $(e, \alpha) \in PA_0(r_e) \subset PA_0(UA_0(x))$. Положим $y = x$, $s_y = s_x$, следовательно, истинен предикат $directly\_access\_own(s_x, s_x, G_0)$, выполнено условие 2 определения 9, и предикат $directly\_can\_share((e, \alpha), x, G_0)$ является истинным.

Второе условие: $op_1 = grant\_right(s_x, r_e, (e, \alpha))$. Следовательно, роль $r_e \in roles_0(s_x) \subset UA_0(x)$, выполняется условие

$$((e, own_r), r_e) \in de\_facto\_actions_0(s_x),$$

и существует роль $r'_e \in R$ такая, что $(e, own_r) \in PA_0(r'_e)$, и либо $r'_e \in \in roles_0(s_x) \subset UA_0(user_0(s_x))$, либо существует субъект-сессия $s_y \in S_0$:

$(s_x, s_y, own_a) \in A_0$, $r'_e \in roles_0(s_y)$ и $r_e \in can\_manage\_rights(roles_0(s_y) \cap$ $\cap AR)$. При этом так как $(e, \alpha) \notin de\_facto\_rights_0(s_x)$, то $\alpha \neq own_r$.

Если $r'_e \in UA_0(user_0(s_x)) = UA_0(x)$, то $(e, own_r) \in PA_0(UA_0(user_0(s_x)))$. Положим $y = x$, $s_y = s_x$. Тогда истинен предикат

$$directly\_access\_own(s_x, s_x, G_0),$$

выполнено условие 2 определения 9, и предикат

$$directly\_can\_share((e, \alpha), x, G_0)$$

является истинным.

Если существует субъект-сессия $s_y \in S_0$: $(s_x, s_y, own_a) \in A_0$, $r'_e \in$ $\in roles_0(s_y)$ и $r_e \in can\_manage\_rights(roles_0(s_y) \cap AR) \cap UA_0(x)$, то положим $y = user_0(s_y)$. Следовательно, выполнено условие 5 определения 7, и истинен предикат $directly\_access\_own(s_x, s_y, G_0)$. Тогда, если $s_y \in N_S \cap$ $\cap S_0$, то выполнено условие 2 определения 9, и является истинным предикат $directly\_can\_share((e, \alpha), x, G_0)$. Если $s_y \in L_S \cap S_0$, то выполнено условие 4 определения 9, и предикат $directly\_can\_share((e, \alpha), x, G_0)$ является истинным.

Третье условие: существует недоверенная субъект-сессия $s_y \in N_S \cap S_0$, $s_y \neq s_x$, и $op_1 = grant\_right(s_y, r_e, (e, \alpha))$. Следовательно, $((e, own_r), r_e) \in$ $\in de\_facto\_actions_0(s_y)$ и существует роль $r'_e \in R$ такая, что $(e, own_r) \in$ $\in PA_0(r'_e)$, и либо

$$r'_e \in roles_0(s_y) \subset UA_0(user_0(s_y)) \text{ и } r_e \in can\_manage\_rights(roles_0(s_y) \cap AR),$$

либо существует субъект-сессия $s'_y \in S_0$: $(s_y, s'_y, own_a) \in A_0$, $r'_e \in roles_0(s'_y)$ и $r_e \in can\_manage\_rights(roles_0(s'_y) \cap AR)$.

Если $r'_e \in UA_0(user_0(s_y))$, то положим $y = user_0(s_y)$. Следовательно, выполнено условие 4 определения 8, и является истинным предикат $directly\_grant\_right(s_y, s_x, G_0)$. Тогда выполнено условие 6 определения 9, и предикат $directly\_can\_share((e, \alpha), x, G_0)$ является истинным.

Если существует субъект-сессия $s'_y \in S_0$: $(s_y, s'_y, own_a) \in A_0$, $r'_e \in$ $\in roles_0(s'_y)$ и $r_e \in can\_manage\_rights(roles_0(s'_y) \cap AR)$, то $s_y \neq s'_y$ и из условия теоремы следует, что $s'_y = s_x$. Следовательно, выполняются условия

$$r'_e \in roles_0(s_x) \subset UA_0(user_0(s_x)) = UA_0(x), \quad (e, own_r) \in PA_0(UA_0(user_0(s_x))).$$

Положим $y = x$. Тогда истинен предикат $directly\_access\_own(s_x, s_x, G_0)$, выполнено условие 2 определения 9, и является истинным предикат $directly\_can\_share((e, \alpha), x, G_0)$.

Второй случай: право доступа $(e, \alpha) \notin PA_1(roles_1(s_x))$, и существуют субъект-сессия $s_y \in S_1$: $(s_x, s_y, own_a) \in A_1$, и роль $r_e \in roles_1(s_y)$:

$(e, \alpha) \in PA_1(r_e)$. Тогда из минимальности N и определения правил преобразования состояний следует, что $s_y \in S_0$ и либо $(s_x, s_y, own_a) \in A_0$ и $(e, \alpha) \notin PA_0(roles_0(s_y))$, либо $(s_x, s_y, own_a) \notin A_0$ и $(e, \alpha) \in PA_0(roles_0(s_y))$. Таким образом, выполняется одно из пяти условий.

Первое условие: $s_y \in N_S \cap S_0$, $(s_x, s_y, own_a) \in A_0$ и $op_1 = take\_role(s_y, r_e)$. Положим $y = user_0(s_y)$. Тогда

$$(e, \alpha) \in PA_0(r_e), \quad r_e \in roles_0(s_y) \subset UA_0(user_0(s_y)) = UA_0(y),$$

истинен предикат $directly\_access\_own(s_x, s_y, G_0)$, выполнено условие 2 определения 9, и предикат $directly\_can\_share((e, \alpha), x, G_0)$ является истинным.

Второе и третье условия, когда $op_1 = grant\_right(s_x, r_e, (e, \alpha))$ или $op_1 = grant\_right(s_y, r_e, (e, \alpha))$, соответственно, рассматриваются аналогично второму и третьему условиям первого случая.

Четвертое условие: $op_1 = control(s_x, s_y, z)$, где $z \in [s_y]$. Тогда $(e, \alpha) \in PA_0(roles_0(s_y))$, $(s_x, s_y, own_a) \notin A_0$, и или $s_x = z$, или $(s_x, z, write_m) \in F_0$. Следовательно, выполняется условие 5 определения 7, и истинен предикат $directly\_access\_own(s_x, s_y, G_0)$. Положим $y = user_0(s_y)$. Тогда, если $s_y \in N_S \cap S_0$, то выполнено условие 2 определения 9, и предикат $directly\_can\_share((e, \alpha), x, G_0)$ является истинным. Если $s_y \in L_S \cap S_0$, то выполнено условие 3 определения 9, и является истинным предикат $directly\_can\_share((e, \alpha), x, G_0)$.

Пятое условие: $op_1 = access\_own(s_x, s_y)$. Тогда $(e, \alpha) \in PA_0(roles_0(s_y))$, $(s_x, s_y, own_a) \notin A_0$ и $(s_y, own_r) \in de\_facto\_rights_0(s_x)$. Следовательно, по условию теоремы $(s_y, own_r) \in PA_0(UA_0(user_0(s_x))) \subset PA_0(UA_0(user_0(s_x))) = PA_0(UA_0(x))$, выполняется условие 5 определения 7, и истинен предикат $directly\_access\_own(s_x, s_y, G_0)$. Положим $y = user_0(s_y)$. Тогда, если $s_y \in N_S \cap S_0$, то выполнено условие 2 определения 9, и предикат $directly\_can\_share((e, \alpha), x, G_0)$ является истинным. Если $s_y \in L_S \cap S_0$, то выполнено условие 3 определения 9, и является истинным предикат $directly\_can\_share((e, \alpha), x, G_0)$.

Таким образом, доказано, что при длине траектории $N = 1$, если истинен предикат $can\_share((e, \alpha), x, G_0)$, то является истинным предикат $directly\_can\_share((e, \alpha), x, G_0)$.

Пусть $N > 1$ и утверждение теоремы верно для всех траекторий длины $l < N$. Докажем, что при длине траектории N, если истинен предикат $can\_share((e, \alpha), x, G_0)$, то истинен $directly\_can\_share((e, \alpha), x, G_0)$.

Из минимальности N следует, что выполняется условие $(e, \alpha) \notin de\_facto\_rights_{N-1}(s_x)$. Возможны два случая.

Первый случай: право доступа $(e, \alpha) \in PA_N(roles_N(s_x))$. Тогда $(e, \alpha) \notin PA_{N-1}(roles_{N-1}(s_x))$ и существует роль $r_e \in roles_N(s_x)$ такая, что $(e, \alpha) \in$

$\in PA_N(r_e)$. Из определения правил преобразования состояний следует, что либо $op_N = take\_role(s_x, r_e)$, либо $op_N = grant\_right(s_x, r_e, (e, \alpha))$, либо существует субъект-сессия $s_y \in N_S \cap S_{N-1}$ и $op_N = grant\_right(s_y, r_e, (e, \alpha))$.

Пусть $op_N = take\_role(s_x, r_e)$. Тогда $r_e \in UA_{N-1}(user_{N-1}(s_x)) = UA_{N-1}(x)$, $(e, \alpha) \in PA_{N-1}(r_e) \subset PA_{N-1}(UA_{N-1}(x))$. По предположению 1 выполняется условие $r_e \in UA_{N-1}(x) = UA_0(x)$, следовательно, $(e, \alpha) \in PA_{N-1}(UA_0(x))$. Таким образом, выполняется одно из двух условий.

Первое условие: право доступа $(e, \alpha) \in PA_0(r_e)$. Тогда $(e, \alpha) \in PA_0(UA_0(x))$. Положим $y = x$. Следовательно, предикат $directly\_access\_own(x, x, G_0)$ является истинным, выполнено условие 1 определения 9, и истинен предикат $directly\_can\_share((e, \alpha), x, G_0)$.

Второе условие: право доступа $(e, \alpha) \notin PA_0(r_e)$. Тогда существует $1 \leqslant M < N$ такое, что возможны две ситуации.

Первая ситуация: субъект-сессия $s_x \in N_S \cap S_{M-1}$ и выполнено $op_M = grant\_right(s_x, r_e, (e, \alpha))$. Тогда выполняется условие $((e, own_r), r_e) \in de\_facto\_actions_{M-1}(s_x)$ и $(e, own_r) \in de\_facto\_rights_{M-1}(s_x)$, при этом так как $(e, \alpha) \notin de\_facto\_rights_{M-1}(s_x)$, то $\alpha \neq own_r$. Следовательно, по определению 6 истинен предикат $can\_share((e, own_r), x, G_0)$ с длиной траектории меньшей N, по предположению индукции истинен предикат $directly\_can\_share((e, own_r), x, G_0)$, и по определению 9 выполнено одно из условий 1–3, 5, 6 его истинности. Если выполняется одно из условий 1, 2, 5, 6, то по определению 9 выполнены соответствующие условия истинности предиката $directly\_can\_share((e, \alpha), x, G_0)$.

Пусть для предиката $directly\_can\_share((e, own_r), x, G_0)$ выполнено условие 3 определения 9. Тогда право доступа $\alpha \neq own_r$, $((e, own_r), r_e) \in de\_facto\_actions_{M-1}(s_x)$, существует роль $r'_e \in R$ такая, что $(e, own_r) \in PA_{M-1}(r'_e)$, и существует субъект-сессия $s_y \in S_{M-1}$: $(s_x, s_y, own_a) \in A_{M-1}$, $r'_e \in roles_{M-1}(s_y)$ и $r_e \in can\_manage\_rights(roles_{M-1}(s_y) \cap AR)$. По условию теоремы, условию 3 определения 9 субъект-сессия s_y является доверенной $s_y \in L_S \cap S_0$, и выполняются условия: $roles_0(s_y) = roles_{M-1}(s_y)$, $(e, own_r) \in PA_0(roles_0(s_y))$, $r_e \in can\_manage\_rights(roles_0(s_y) \cap AR) \cap UA_0(x)$, истинен предикат $directly\_access\_own(x, s_y, G_0)$. Положим $y = user_0(s_y)$. Следовательно, выполняется условие 4 определения 9, и предикат $directly\_can\_share((e, \alpha), x, G_0)$ является истинным.

Вторая ситуация: существует недоверенная субъект-сессия $s_y \in N_S \cap S_{M-1}$, $s_y \neq s_x$, и $op_M = grant\_right(s_y, r_e, (e, \alpha))$. Тогда выполняется условие $((e, own_r), r_e) \in de\_facto\_actions_{M-1}(s_y)$ и право доступа $(e, own_r) \in de\_facto\_rights_{M-1}(s_y)$. Положим $y = user_{M-1}(s_y)$, тогда по определению 6 истинен предикат $can\_share((e, own_r), y, G_0)$ с длиной траектории меньшей N, по предположению индукции истинен предикат $directly\_can\_share((e, own_r), y, G_0)$. Следовательно, по условию теоре-

мы и по определению 9 выполнено одно из условий 1, 2, 5, 6 его истинности. Если выполняется одно из условий 1, 2, то по определению 9 выполнены соответствующие условия истинности предикатов $directly\_can\_share((e, own_r), x, G_0)$ и $directly\_can\_share((e, \alpha), x, G_0)$. Если выполнено одно из условий 5, 6 определения 9, то либо $(e, own_r) \in \in PA_0(UA_0(x))$, либо $(e, own_r) \in PA_0(UA_0(y))$. Если $(e, own_r) \in PA_0(UA_0(x))$, то по условию 1 определения 9 предикат $directly\_can\_share((e, \alpha), x, G_0)$ является истинным. Если $(e, own_r) \in PA_0(UA_0(y))$, то так как $((e, own_r), r_e) \in \in de\_facto\_actions_{M-1}(s_y)$, то $r_e \in can\_manage\_rights(AUA_0(y)) \cap UA_0(x)$. Следовательно, выполняется условие 1 определения 8, истинен предикат $directly\_grant\_right(y, x, G_0)$, и по условию 5 определения 9 предикат $directly\_can\_share((e, \alpha), x, G_0)$ является истинным.

Если $op_N = grant\_right(s_x, r_e, (e, \alpha))$, или существует субъект-сессия $s_y \in N_S \cap S_{N-1}$ и $op_N = grant\_right(s_y, r_e, (e, \alpha))$, то выполняется соотношение $r_e \in UA_{N-1}(user_{N-1}(s_x)) = UA_{N-1}(x) = UA_0(x)$, и аналогично доказывается, что предикат $directly\_can\_share((e, \alpha), x, G_0)$ является истинным.

Обоснование шага индукции для второго случая, когда право доступа $(e, \alpha) \notin PA_N(roles_N(s_x))$, осуществляется с применением техники доказательства аналогичной использованной в первом случае и в случае $N = 1$.

Следовательно, доказан шаг индукции: при длине траектории N, если истинен предикат $can\_share((e, \alpha), x, G_0)$, то истинен предикат $directly\_can\_share((e, \alpha), x, G_0)$. Доказательство необходимости выполнения условия теоремы для истинности предиката $can\_share((e, \alpha), x, G_0)$ выполнено.

Теорема доказана. □

Таким образом, в теореме 1 для случая, когда для передачи права доступа непосредственно взаимодействуют только две субъект-сессии, обоснованы необходимые и достаточные условия, при выполнении которых субъект-сессия может получить роль или фактическую роли, обладающую заданным правом доступа к сущности.

В дальнейшем предполагается применение БР ДП-модели в КС с РУД для анализа условий передачи прав доступа ролей для случая произвольного числа субъект-сессий, а также для анализа условий возникновения информационных потоков по памяти или по времени.

Литература

[1] *Девянин П. Н.* Анализ безопасности управления доступом и информационными потоками в компьютерных системах. М.: Радио и связь, 2006, 176 с.

[2] *Девянин П. Н.* Модели безопасности компьютерных систем: Учеб. пособие для студ. высш. учеб. заведений. М.: Издательский центр «Академия», 2005, 144 с.

[3] *Bishop M.* Computer Security: Art and Science. ISBN 0-201-44099-7, 2002, 1084 p.

[4] *McLean J., John D.* The Specification and Modeling of Computer Security // Computer, 1990, vol. 23, no. 1.

[5] *Sandhu R.* Role-Based Access Control. Advanced in Computers, vol. 46, Academic Press, 1998.

Подход к управлению настройками механизмов безопасности в дистрибутивах ОС Linux

К. А. Шапченко, О. О. Андреев

Реализация положений политики безопасности информационно-вычислительной системы, которая понимается далее как совокупность правил, направленных на безопасное использование информации, технологий и других ресурсов такой системы, зачастую требует проведения ряда действий по изменению настроек программно-технических механизмов, задействованных в обеспечении информационной безопасности. В связи с существенно динамическим характером таких операций возникает необходимость организации проверки выполнения требований по обеспечению информационной безопасности в процессе настройки указанных механизмов. В настоящей работе на уровне постановки задач и выделения общих положений их решения предлагается подход к созданию и использованию программных средств, поддерживающих процесс такой проверки. Задача рассматривается для случая поддержки политики информационной безопасности путем изменения настроек механизмов логического разграничения доступа в операционных системах на основе ядра Linux.

В контексте требований существующих нормативно-правовых документов следует отметить, что рассматриваемая задача непосредственно связана с управлением функциями безопасности, отдельные требования к которому представлены в ГОСТ Р ИСО/МЭК 15408 «Общие критерии», а именно — в описании класса требований FMT («Управление безопасностью»). Решение рассматриваемой задачи позволит не только ограничить изменение настроек механизмов безопасности (включая «атрибуты безопасности» в терминологии стандарта), но и предоставит возможность организовать проверку выполнения ряда требований по безопасности в состоянии защищаемой системы до и после производимых изменений.

Анализ выполнения требований по безопасности в процессе изменения настроек поддерживающих их механизмов особенно важен при изменении этих требований, включая, например, их пересмотр и уточнение. В случае, если подобные требования статичны, что характерно для некоторых упрощенных (в плане их функциональных возможностей) информационно-вычислительных систем, можно реализовать режим поддержки их защи-

ты без изменений настроек механизмов безопасности. К таким системам можно отнести, например, те, в которых не производится создания новых субъектов и объектов доступа и аналогичных операций, либо те, в которых подобные действия не влияют на выполнение требований по безопасности, и эти свойства заложены в ее архитектуре и реализации. Однако даже в таких системах при пересмотре требований по безопасности возникает необходимость изменения настроек механизмов безопасности, задаваемых «статично». В процессе такого изменения представляется целесообразным удостовериться, какой эффект на защищаемой системе оно имеет, а также организовать проверку выполнения заданных требований. Важность такой проверки возрастает для информационно-вычислительных систем, в которых операции по изменению настроек механизмов безопасности являются частыми.

Необходимость подобной проверки также подтверждается существенной ролью человеческого фактора в управлении настройками механизмов безопасности, которое зачастую производится в интерактивном режиме. Традиционные программно-технические механизмы обеспечения безопасности позволяют ограничить доступ оператора (в том числе выступающего в роли администратора безопасности) к функциям по настройке параметров информационно-вычислительной системы, включая имеющие отношение к механизмам обеспечения ее безопасности. Вместе с тем, такого ограничения в ряде случаев может оказаться недостаточно. Это обусловлено тем фактом, что, как правило, в рамках действий, разрешенных администратору безопасности, в принудительном режиме и без оценки возможных последствий может быть произведено изменение настроек программно-технических механизмов, приводящее к нарушению заданных требований по безопасности. В качестве примера подобных действий можно привести изменение настроек механизмов разграничения доступа в операционных системах. Как правило, администратор безопасности (а в случае использования дискреционных моделей — и пользователь-владелец) может как запретить доступ к некоторому объекту, так и разрешить его, что в свою очередь может привести к противоположным последствиям в зависимости от требований, установленных политикой безопасности. Отличительной особенностью традиционных средств управления настройками механизмов безопасности является отсутствие возможности предоставить оператору необходимые сведения о возможных последствиях проводимых изменений, а также интерпретации таких последствий согласно принятым требованиям по обеспечению безопасности.

Следует упомянуть один из традиционных подходов к проверке требований по безопасности с помощью внешнего по отношению к защищаемой системе аудита настроек. При таком подходе происходит сбор параметров

функционирования защищаемой системы (включая настройки механизмов безопасности) и их последующий анализ, проводимый независимо от работы системы. Соответственно, теряется возможность оперативного реагирования на нарушения конфигурации механизмов обеспечения безопасности. Тем не менее, для реализации подобного подхода создаются модели и средства проверки ряда классов логических свойств, которые с успехом могут быть использованы в подходе, предлагаемом далее.

Отметим ряд особенностей решаемой задачи. Политика информационной безопасности подконтрольной системы, как правило, не имеет формального описания. Указанное обстоятельство усложняет решение задачи уже на этапе ее постановки. Причина в том, что сложно оценить, какие именно свойства и каким образом могут быть рассмотрены в процессе автоматизированной проверки. Тем не менее, функционирование и настройки ряда механизмов безопасности (например, механизмов логического разграничения доступа) могут быть формализованы в виде набора математических моделей, отражающих действительное состояние настроек защищаемой системы. Более того, для таких моделей существуют подходы к автоматической либо автоматизированной проверке отдельных классов свойств [1, 2, 3, 4]. В ряде случаев (например, для сложно формализуемых свойств) необходимо непосредственное участие человека (аудитора) в проверке выполнения требований. Вместе с тем, зачастую, при наличии достаточно полной формальной модели, описывающей такие свойства, возможно проведение автоматического анализа.

Следует отметить сложность проведения анализа настроек механизмов безопасности. Она обусловлена несколькими предпосылками, к числу которых относятся:

- большой объем анализируемой информации, в том числе моделей может быть несколько, и каждая из них может иметь достаточно большое число подлежащих анализу параметров;
- как правило, в сложно организованных системах возникает необходимость интеграции настроек нескольких разных механизмов безопасности в единую модель;
- с точки зрения необходимости поддержки режима интерактивного анализа состояния настроек выбор способа визуально удобного представления необходимой для этого информации затруднен, так как цели анализа недостаточно формализованы.

В случае, если для моделей, описывающих используемые механизмы безопасности, возможна автоматизация проверки некоторых заданных свойств, возникает отдельная задача спецификации таких свойств. Для ее решения необходимо проведение предварительного исследования модели.

Предлагаемый подход заключается в реализации дополнительного программного механизма, встраиваемого в процесс изменения настроек механизмов обеспечения безопасности таким образом, что подобное изменение инициирует проверку выполнения требований по безопасности. Такое программное решение представляет собой расширение паттернов проектирования (согласно [5]) «монитор обращений» и «единая точка входа» по отношению к управлению настройками безопасности. Таким образом реализуется единообразие при обращении к функциям по изменению настроек.

Кроме интеграции со средствами проверки выполнения требований по безопасности, в качестве дополнительных функциональных требований к работе подобного механизма можно отнести следующие:

- инкрементальный анализ — процесс исследования настроек механизмов безопасности, который, в первую очередь, нацелен на произведенные изменения;
- факторизация настроек для упрощения (вследствие меньшего объема исследуемой модели) спецификации свойств безопасности, их проверки и интерпретации результатов;
- комплексное изменение настроек (в отличие от изменения их по одной) с проведением анализа последствий подобных крупных изменений параметров функционирования защищаемой системы;
- интерактивная визуализация настроек механизмов безопасности и изменений в них.

Кроме того, отметим возможность расширения такого подхода в силу его универсальности не только на механизмы логического разграничения доступа в операционных системах, но и на другие механизмы обеспечения безопасности, в том числе — в приложениях и сервисах прикладного уровня. Для этого необходимо обеспечить применение рассматриваемого подхода к интегрированному набору программных средств, как системного, так и прикладного уровня. Такие действия целесообразно проводить в рамках создания отдельного дистрибутива операционной системы.

В рамках предлагаемого подхода интерес представляет создание языка промежуточного уровня для описания настроек механизмов безопасности и с его помощью формализация общей модели для защищаемой целевой системы или отдельных ее компонентов. Такой язык призван занять нишу между высокоуровневым, неформальным, как правило, словесным описанием политики безопасности и низкоуровневыми настройками отдельных программных средств. На настоящее время существует ряд исследований, направленных на разработку языков описания отдельных требова-

ний политик безопасности и настроек механизмов безопасности [6, 7, 8, 9]. В рамках предлагаемого подхода целесообразно за основу принять один из разработанных языков, изменив, по возможности, его с целью обеспечить соответствие поставленным задачам анализа и изменения конфигурации механизмов безопасности.

Необходимость создания и использования отдельного языка объясняется гетерогенностью механизмов обеспечения безопасности, различием языков, описывающих настройки таких механизмов, а также математических моделей, на которых основано их функционирование. Таким образом, для проведения анализа конфигурации средств обеспечения безопасности защищаемой информационно-вычислительной системы на предмет их соответствия заданным требованиям или для изменения настроек безопасности для соответствия таким требованиям необходимо:

- выделение из общего набора настроек тех, которые относятся к проверяемым требованиям;
- согласование и проверка непротиворечивости настроек, используемых в нескольких различных механизмах безопасности.

В качестве примера использования языка промежуточного уровня рассмотрим проверку свойства «Учетная запись администратора системы должна быть защищена паролем». Такое свойство в Unix-подобных операционных системах соответствует следующим настройкам:

- система аутентификации содержит непустой пароль для пользователя `root`, проверка данной настройки зависит от используемой системы аутентификации и может включать проверку записей в локальных файлах, а также в глобальной сетевой системе конфигурации;
- система передачи прав другим пользователям не должна разрешать им выполнять программы без ввода пароля.

Отсюда следует, что даже для проверки простого правила необходимо проверить конфигурацию как минимум двух различных механизмов безопасности, аудитор при этом должен знать синтаксис и семантику конфигурации каждого из этих средств, а также, возможно, других сервисов, используемых этими механизмами.

Язык промежуточного уровня можно использовать как для описания настроек безопасности, так и для задания проверяемых свойств. Одним из способов использования предлагаемого к разработке языка является внедрение механизмов безопасности, настройка которых производится с помощью общего языка промежуточного уровня, в программные средства

целевой системы. Такой способ позволяет задавать настройки безопасности унифицированным способом напрямую во всех программных компонентах системы, что может упростить выделение настроек, подлежащих анализу, и их изменение. В то же время, данный способ обладает рядом недостатков, к числу которых относятся следующие.

- Он может быть реализован либо в виде модификации кода программных средств, либо в виде создания дополнительных модулей безопасности, в том случае, если архитектура программных комплексов поддерживает подключение таких модулей.
- Процесс модификации программных средств весьма трудоемок, в том числе, из-за немалого объема дублирующегося между несколькими средствами кода, а также из-за необходимости постоянного обновления при выходе новых версий модифицированных программных средств.
- Внедрение большого количества необходимых изменений в код используемых программных средств усложняет аудит программного кода на предмет наличия уязвимостей.

С учетом изложенных недостатков можно констатировать, что существенная модификация программных средств будет неэффективной, следовательно, данный способ представляется малоперспективным.

Другим подходом к задаче использования языка промежуточного уровня для описания настроек механизмов безопасности целевой системы является реализация преобразования правил, записанных на языке промежуточного уровня, в настройки механизмов безопасности, без изменения самих механизмов и моделей их работы, а также обратного преобразования из таких настроек в набор правил, записанных на таком языке. Подобный подход представляется наиболее перспективным по следующим причинам:

- требуемые изменения используемых программных средств менее трудоемки и менее подвержены ошибкам;
- возможно проведение плавного, «незаметного» внедрения таких средств обеспечения безопасности в уже существующие системы защиты с сохранением существующих настроек.

Таким образом, в общем случае не возникает необходимости администратору безопасности и аудитору безопасности целевой системы изучать конфигурацию отдельных механизмов безопасности, присутствующих в системе. Появляется лишь необходимость описать настройки на языке промежуточного уровня. При этом низкоуровневые, частные детали конфигура-

ции отдельных средств при преобразовании в промежуточное представление (то есть, представление на языке промежуточного уровня) трансформируется в общие конструкции языка описания. Вместе с тем, при необходимости изменения отдельных настроек, например, в случае с установкой стороннего программного пакета, такое изменение будет отражено в виде изменения промежуточного представления и в наглядной форме представлено аудитору или администратору безопасности. Данный подход ранее в основном применялся к узкому кругу задач по настройке и проверке непротиворечивости настроек межсетевых экранов [10, 11].

Для реализации подобного подхода потребуется создание нескольких независимых инструментальных средств, в числе которых:

- транслятор или набор трансляторов из конструкций языка промежуточного уровня в низкоуровневые настройки отдельных механизмов безопасности;
- транслятор или набор трансляторов из низкоуровневых настроек механизмов безопасности в язык промежуточного уровня;
- средство проверки и обнаружения конфликтов в промежуточном представлении;
- средство для визуализации промежуточного представления и его изменений.

Средство проверки и обнаружения конфликтов может применяться как для обнаружения противоречий между настройками отдельных средств обеспечения безопасности, так и для обнаружения противоречий вносимых администратором безопасности изменений с заранее заданными требованиями, которым должна удовлетворять целевая система.

Литература

[1] *Андреев О. О.* Язык описания моделей разграничения доступа и его реализация в ядре операционной системы Linux // Математика и безопасность информационных технологий. (Материалы конференции в МГУ 2–3 ноября 2005 г.) М.: МЦНМО, 2006, с. 305–321.

[2] *Шапченко К. А.* К вопросу о средствах ОС Linux для управления доступом при использовании ролевых политик безопасности // Математика и безопасность информационных технологий. (Материалы конференции в МГУ 2–3 ноября 2005 г.) М.: МЦНМО, 2006, с. 257–281.

[3] *Васенин В. А.*, *Шапченко К. А.*, *Андреев О. О.* Математические модели и механизмы логического разграничения доступа в операционной системе

Linux: текущее состояние и перспективы развития // Математика и безопасность информационных технологий. (Материалы конференции в МГУ 25–26 октября 2006 г.) М.: МЦНМО, 2007, с. 159–171.

[4] *Guttman J. D.*, *Herzog A. L.*, *Ramsdell J. D.*, *Skorupka C. W.* Verifying information flow goals in Security-Enhanced Linux // Journal of Computer Security. 2005. V. 13, №. 1. P. 115–134.

[5] *Schumacher M.*, *Fernandez-Buglioni E.*, *Hybertson D.*, *Buschmann F.*, *Sommerlad P.* Security Patterns: Integrating Security and Systems Engineering // Wiley Software Patterns Series. John Wiley & Sons, March 2006.

[6] *Jajodia S.*, *Samarati P.*, *Subramanian V.* A logical language for expressing authorizations // Proceedings of the 1997 IEEE Symposium on Security and Privacy, 1997, p. 31–42.

[7] *Cholvy L.*, *Cuppens F.* Analyzing consistency of security policies // Proceedings of the 1997 IEEE Symposium on Security and Privacy, 1997, p. 103–112.

[8] *Damianou N.*, *Dulay N.*, *Lupu E.*, *Sloman M.* The Ponder policy specification language // Proceedings of Policies for Distributed Systems and Networks, 2001, p. 18–38.

[9] OASIS XACML Technical Committee. XACML 2.0 Core: eXtensible Access Control Markup Language (XACML) Version 2.0., 2007.

[10] *Al-Shaer E.*, *Hamed H.* Discovery of policy anomalies in distributed firewalls // Proceedings of IEEE Infocomm, 2004.

[11] *Bartal Y.*, *Mayer A.*, *Nissim K.*, *Wool A.* Firmato: A novel firewall management toolkit // Proceedings of the 1999 IEEE Symposium on Security and Privacy, 1999.

Конфигурируемая модульная система мониторинга поведения транспортного протокола на уровне ядра операционной системы

В. А. Пономарев, О. Ю. Богоявленская,
Богоявленский Ю. А.

1. Введение

В работе [1] описана система `GetTCP`, которая позволяет получать локализованные в ядре операционной системы (ОС) Linux данные о состоянии и динамике внутренних переменных, отражающих поведение соединений TCP. Кроме того, при разработке `GetTCP` были приняты меры по уменьшению накладных расходов, возникающих при работе системы. Указанные особенности позволяют использовать систему `GetTCP` для ряда задач, решение которых невозможно с помощью общеизвестных средств мониторинга поведения TCP. Так, система `GetTCP` применялась для сбора данных, необходимых для проверки адекватности модели TCP [2]. С точки зрения безопасности такая система предоставляет возможность интегрировать в ОС все возможные функции защиты на уровне TCP.

Вместе с тем, первоначальный вариант системы `GetTCP` обладал рядом недостатков, описанных в [3]. Наиболее существенные из этих недостатков — отсутствие библиотеки пользовательского уровня, обеспечивающей доступ к возможностям системы, а также необходимость перекомпиляции загружаемого модуля ядра при любом изменении набора перехватываемых данных.

В статье описана новая версия системы `GetTCP`, в которой были устранены указанные недостатки. Также рассматриваются примеры использования новой версии.

2. Получение управления при передаче данных

Возможность перехвата внутренних данных ядра ОС — ключевая особенность системы `GetTCP`. Очевидно, что для доступа к таким данным

часть системы должна действовать в адресном пространстве ядра и получать управление процессором при отправлении каждого сегмента TCP. Рассмотрим варианты, которыми можно реализовать получение управления подсистемой перехвата при отправке сегмента.

В первоначальной реализации системы `GetTCP` применялся простой способ без использования каких-либо готовых механизмов ядра. В исходный код ядра ОС Linux, осуществляющий передачу сегмента TCP, непосредственно перед обращением к функции отправки данных сетевого уровня добавлялся код, передающий управление системе `GetTCP`.

Указанный подход требует модификации исходного кода ядра, перекомпиляции ядра и перезагрузки вычислительной системы для начала работы. В то же время в современном ядре ОС Linux существует возможность динамической установки «контрольных точек» и последующего вызова заданной при установке «контрольной точки» функции с помощью механизма `KProbes` [4]. Механизм предоставляет несколько разновидностей «контрольных точек», была исследована одна из этих разновидностей — `jprobe`.

В связи с тем, что величина задержек, вносимых системой `GetTCP` в передачу сегмента данных, является важнейшим элементом спецификации системы, были проведены эксперименты по измерению времени, необходимого для получения управления с использованием механизма `jprobe`. Измерения проводились на той же вычислительной системе, которая использовалась для экспериментов в [1] (ПЭВМ на основе процессора Celeron с тактовой частотой 3ГГц). Выяснилось, что получение управления занимает минимум 3185 тактов процессора, что составляет около 1 мкс. Учитывая то, что в старой версии системы `GetTCP` общие накладные расходы на обработку сегмента в 94.02% случаев не превышали 0.3 мкс, было принято решение отказаться от использования `KProbes`.

Любопытно, что при проведении измерений пришлось учитывать влияние попадания или непопадания кода в кэш-память процессора (при непопадании время выполнения тестовой функции увеличивалось на несколько порядков). По этой причине проводилось большое количество (сотни тысяч) измерений с минимальными временны́ми интервалами между измерениями, чтобы код тестовой функции не успевал покинуть кэш процессора. Для минимизации этих временны́х интервалов, в свою очередь, было реализовано программное обеспечение (ПО) для тестирования на языке Си. Первоначально использовался язык Perl, однако выяснилось, что при этом код тестовой функции успевает покинуть кэш процессора и результаты измерений получаются недостоверными.

Другой механизм — kernel markers [5], стал доступен сравнительно недавно и был включен в «официальное» ядро ОС Linux в версии 2.6.24,

которая вышла 24 января 2008 года (именно в то время, когда проводились описанные выше измерения для механизма `KProbes`). Kernel markers представляют собой статические «маркеры», заранее включенные в исходный код ядра ОС Linux (в отличие от динамических контрольных точек механизма `KProbes`, которые устанавливаются во время выполнения). Такой подход имеет свои преимущества и области применения. Статические «маркеры» сопровождаются вместе с исходным кодом ядра ОС Linux, и, следовательно, всегда согласованы с ним. Кроме того, разработчиками kernel markers были приняты специальные меры по уменьшению накладных расходов при использовании этого механизма.

В экспериментальное ПО, разработанное ранее для `KProbes`, было добавлено измерение накладных расходов механизма kernel markers. Выяснилось, что для получение управления требуется 138 тактов процессора, т.е. около 0.05 мкс (измерения проводились на той же ПЭВМ на основе процессора Celeron с тактовой частотой 3ГГц). При этом измерялось непосредственно время, затрачиваемое на вызов функции из «маркера», в отличие от подхода, примененного в [6], где измерялись косвенные «макропараметры» (время, затрачиваемое на монтирование-размонтирование тома ext3).

Учитывая указанные выше обстоятельства, было принято решение об использовании для новой версии `GetTCP` механизма kernel markers. Необходимость внесения изменений в исходный код ядра и последующей перекомпиляции ядра при использовании статических «маркеров» были признаны неизбежными, т.к. другие доступные механизмы (`jprobe`) приводят к недопустимому увеличению задержки при отправке сегмента TCP.

Для размещения «маркера» в ядре используется следующая конструкция:

```
trace_mark(gettcp_tcp_probe, "sk_buff %p integer %d", skb, id);
```

Здесь `gettcp_tcp_probe` — имя «маркера», `skb` — указатель на внутреннюю структуру данных ядра ОС Linux (socket buffer), а `id` позволяет однозначно определить строку кода, из которой была вызвана передача сегмента TCP, и, следовательно, условия, в которых происходила передача сегмента (например, состояние алгоритма избежания перегрузок TCP).

3. Архитектура новой версии `GetTCP`

Существенным образом была переработана внутренняя архитектура части системы `GetTCP`, находящейся в ядре ОС. Эта часть была разделена на четыре подсистемы: управления, передачи данных на пользовательский

уровень, получения необходимых данных, измерения накладных расходов, возникающих при работе. Для каждой из этих подсистем доступны три обязательные функции: инициализация подсистемы, сообщение о состоянии подсистемы (для просмотра пользователем), завершение работы подсистемы. Также доступны специфичные для каждой из подсистем функции (изменение конфигурации, принудительная смена под-буфера и т. п.).

Подсистема управления осуществляет получение команд от пользовательского уровня и их обработку. Команды в новой версии `GetTCP` представляют собой строки, состоящие из обязательной команды и одного или нескольких параметров, разделенных пробельными символами (см. пример ниже).

Реализация текстовых сообщений о результате выполнения команд привела бы к чрезмерному усложнению и увеличению кода, работающего в адресном пространстве ядра, что нежелательно. По этой причине было принято решение о компромиссе — реализация интерфейса посредством текстовых команд с ответами в виде кода возврата системного вызова `write`, с помощью которого подается команда. Таким образом, ошибка при выполнении команды выглядит как ошибка записи в управляющий файл:

```
# echo недопустимая команда > /sys/kernel/debug/gettcp/control
bash: echo: write error: Invalid argument
#
```

Отсутствие ошибки выглядит как успешная запись в управляющий файл:

```
# echo chan create 131072 16 > /sys/kernel/debug/gettcp/control
#
```

При обработке команд учитывается состояние, в котором находится система `GetTCP` в момент поступления команды. Например, невозможно включить перехват данных, если не задана конфигурация перехвата, или не создан канал передачи данных на пользовательский уровень.

Для реализации подобных ограничений используется конечный автомат, действующий в четырехмерном пространстве состояний, каждая точка которого может быть описана как (*chan*, *conf*, *prof*, *probe*), где *chan* — состояние подсистемы передачи данных на пользовательский уровень, *conf* — состояние конфигурации перехватываемых данных, *prof* — состояние подсистемы измерения накладных расходов, *probe* — состояние подсистемы получения необходимых данных. Множество допустимых состояний, переходов между ними, а также команд, вызывающих эти переходы, описано в специальной структуре данных. Эта структура данных при

необходимости легко может быть изменена и дополнена. Таким образом, обеспечивается возможность легкого расширения и модификации системы команд `GetTCP`.

Подсистема передачи данных на пользовательский уровень по сравнению с предыдущей версией `GetTCP` изменилась незначительно. Для количества заполненных и обработанных под-буферов в новой версии используются строки, а не двоичные значения. За время, прошедшее с публикации [1], механизм эффективной передачи данных из ядра на пользовательский уровень из файловой системы RelayFS трансформировался в более общий программный интерфейс relay API, для организации файлового интерфейса используется файловая система debugfs [7].

Подсистема получения данных была реализована с помощью описанного выше механизма «маркеров». Появилась также возможность динамического управления набором данных, которые необходимо перехватывать.

Подсистема измерения накладных расходов также была реализована с помощью механизма kernel markers («маркеры» находятся в коде подсистемы получения данных). Основное отличие от предыдущей версии — возможность динамического управления конфигурацией (в новой версии можно указывать ожидаемые диапазоны измерений и количество участков гистограммы).

Значительные изменения произошли в части `GetTCP`, действующей в пользовательском адресном пространстве. Ранее эта часть фактически являлась прототипом и состояла из одной программы, сохраняющей перехваченные данные в файл для последующей обработки. В новой версии `GetTCP` реализована библиотека `libgettcp`, предоставляющая функции для создания и уничтожение канала передачи данных relay API, управления конфигурацией перехвата, запуска и остановки измерения накладных расходов, запуска и остановки перехвата, получения перехваченных данных. Указанную библиотеку можно применять как в оригинальных программах, нуждающихся в возможностях, которые предоставляет только система `GetTCP`, так и в уже существующих утилитах перехвата и обработки сетевого трафика. Для проверки возможности применения библиотеки `libgettcp` в уже существующих программах, а также для тестирования работы библиотеки была использована утилита `fprobe-ulog` [9], служащая для перехвата сетевого трафика с последующим экспортом в формате NetFlow [10]. Указанная утилита была модифицирована для использования библиотеки `libgettcp`. Объем модификаций оказался незначительным (размер полученного diff-файла составил 5.8 кБ). Проведенные эксперименты показали работоспособность полученной модификации утилиты `fprobe-ulog`, а также отсутствие потерь данных при перехвате и формировании потоков NetFlow.

В дополнение к описанным выше изменениям в код всех частей системы `GetTCP` были внесены исправления для работы на 64-битных платформах. Также была проверена компиляция и работа новой версии `GetTCP` на многоядерных платформах (AMD Athlon X2, Intel Core 2 Duo).

В настоящее время ведется работа по улучшению временны́х характеристик подсистемы перехвата и подготовке пользовательской документации. Также разрабатывается библиотека для обеспечения возможности хранения конфигурации системы `GetTCP` в отдельном файле с Си-подобным синтаксисом.

4. Эксперимент по определению реактивности

Для новой версии `GetTCP` был повторен эксперимент по определению задержек, вносимых системой в передачу сегментов TCP. Измерения проводились на следующей аппаратно-программной платформе: ПЭВМ на основе процессора Celeron с тактовой частотой 3ГГц, оперативная память объемом 1.5Гб, интегрированная сетевая карта SiS900 10/100 Мбит/с, локальная сеть Ethernet 100 Мбит/с, ОС Linux, ядро 2.6.25.11 (дистрибутив openSUSE 11.0). Изменения аппаратной части платформы по сравнению с 2006 г. незначительны (объем оперативной памяти увеличен с 512Мб до 1.5Гб). Наиболее значительно изменилась программная часть (ядро 2.6.25 вместо 2.6.16).

Для оценки задержки, вносимой `GetTCP` в передачу сегмента, был проведен эксперимент, аналогичный описанному в [1]. С помощью утилиты Iperf [8] передавался максимально возможный объем данных за фиксированный интервал времени (120 секунд) при фиксированном размере сегмента. Для имитации наиболее неблагоприятных условий работы (минимальное время между отправками сегментов данных) был выбран размер сегмента 50 байт.

Полученные результаты представлены в таблице 1 и на гистограмме (рис. 1) распределения задержек, вносимых `GetTCP` в передачу сегмента TCP (ось ординат в логарифмическом масштабе). Для большинства сегментов (94%) задержка не превышает 1 мкс.

5. Заключение

В статье представлена новая версия системы `GetTCP`, обладающая рядом преимуществ по сравнению с предыдущей версией. При реализации использованы системные механизмы, обеспечивающие малое влияние процесса мониторинга на временны́е характеристики алгоритмов TCP ядра.

Таблица 1. Задержки, вносимые GetTCP в передачу сегмента TCP

Интервал, мкс	Количество сегментов, %
(0, 0.2]	2.5
(0.2, 0.3]	65.2
(0.3, 0.4]	11.6
(0.4, 0.5]	8.4
(0.5, 0.6]	3.4
(0.6, 0.7]	1.8
(0.7, 0.8]	0.7
(0.8, 0.9]	0.3
(0.9, 1]	0.1
> 1	< 6.1

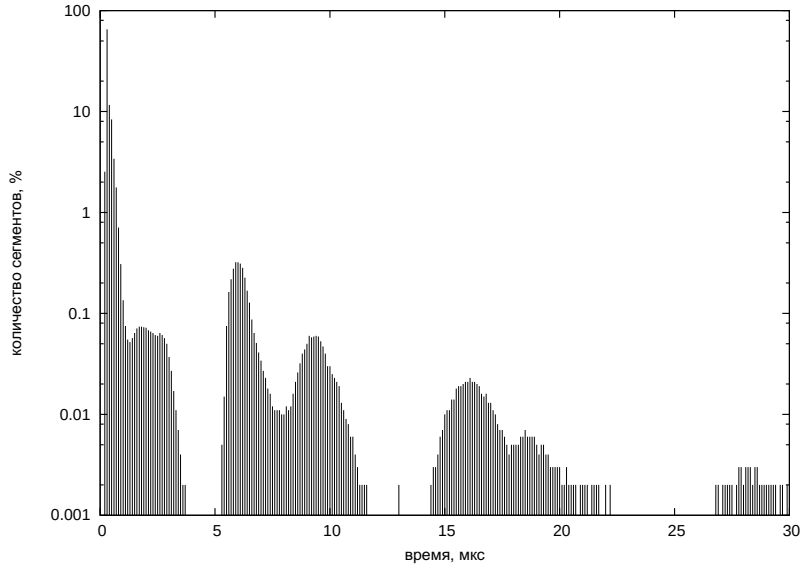


Рис. 1. Гистограмма распределения задержек, вносимых GetTCP в передачу сегмента TCP

Вместе с тем, при реализации новых возможностей (таких, как динамическое изменение конфигурации) не удалось избежать незначительного увеличения вносимых при перехвате задержек. В ближайшее время планиру-

ется проведение экспериментов по определению временны́х характеристик `GetTCP` на более современном оборудовании (процессор Intel Xeon E5420) и более высоких скоростях передачи данных (1 Гбит/с). Также ведется работа по трансформации системы в свободный программный продукт.

Литература

[1] *Пономарев В. А.*, *Богоявленская О. Ю.*, *Богоявленский Ю. А.*, Система мониторинга поведения транспортного протокола на уровне ядра операционной системы, материалы Второй международной научной конференции по проблемам безопасности и противодействия терроризму, стр. 349-357

[2] *Пономарев В. А.*, *Богоявленская О. Ю.*, *Богоявленский Ю. А.*, Проверка адекватности и модификация модели случайного потока, генерируемого транспортным протоколом TCP в сети передачи данных, труды международного семинара «Распределенные компьютерные и телекоммуникационные сети: теория и приложения (DCCN 2007)», том 2, стр. 59—64

[3] *Пономарев В. А.*, *Ковалев В. Н.*, *Богоявленская О. Ю.*, *Богоявленский Ю. А.*, Расширение функций системы мониторинга поведения транспортного протокола на уровне ядра ОС Linux, Труды XIV Всероссийской научно-методической конференции «Телематика 2007», том 1, стр. 101—102

[4] An introduction to KProbes. `http://lwn.net/Articles/132196/`.

[5] Kernel markers. `http://lwn.net/Articles/245671/`.

[6] Measuring Kernel Marker Overhead. `http://kerneltrap.org/Linux/Measuring_Kernel_Marker_Overhead`.

[7] Debugfs. `http://lwn.net/Articles/115405/`.

[8] The TCP/UDP Bandwidth Measurement Tool (iperf). `http://dast.nlanr.net/Projects/Iperf`.

[9] NetFlow probes: fprobe and fprobe-ulog. `http://fprobe.sourceforge.net/`.

[10] Cisco IOS NetFlow Introduction. `http://www.cisco.com/go/netflow`.

О криптографии с открытым ключом на основе задачи разложения языков

С. А. Афонин

Современные криптосистемы с открытым ключом, которые широко применяются на практике, основаны на задачах разложения чисел на множители. Данная задача считается вычислительно сложной, однако ее точный класс сложности неизвестен (предполагается, что она относится к классу NP). В целях повышения криптографической стойкости криптосистем в литературе предлагались различные криптосхемы с открытым ключом, основанные на доказано сложных задачах, в комбинаторных NP-полных задачахили алгоритмических задачах алгебры. Одной из основных проблем при создании криптосистемы с открытым ключом является сложность выбора конкретного экземпляра рассматриваемой вычислительно сложной задачи. Например, если рассматривать NP-полную задачу о рюкзаке, то далеко не каждый выбор весовых коэффициентов порождает действительно вычислительно сложную задачу, хотя универсальный алгоритм, который эффективно решает любую задачу данного типа, неизвестен. В системах с открытым ключом злоумышленник пытается решить один единственный экземпляр задачи, а не любую задачу данного типа.

В данной работе рассматривается возможность построения криптосистемы с открытым ключом на основе задач разложения регулярных языков. Такой выбор связан с высокой алгоритмической сложностью задачи разложения (известные алгоритмы имеют двойную экспоненциальную сложность) и, тем фактом что в настоящее время неизвестно нетривиальных систем языков, допускающих эффективной решение данной задачи. Последнее обстоятельство позволяет надеяться, что при генерации ключей в данном методе можно выбирать произвольные языки с достаточно большим числом состояний соответствующих автоматов.

1. Основные определения

Алфавитом называется конечное непустое множество, элементы которого называются *символами* или *буквами*. Конечная последовательность символов алфавита Σ называется *словом* в этом алфавите. Количество

элементов этой последовательности называется *длиной слова*. Слово, не содержащее ни одного символа, называется *пустым* и обозначается ε. Произвольное множество слов в алфавите называется *языком*. *Конкатенацией* языков L_1 и L_2 называется язык $L_1L_2 = \{w_1w_2 \mid w_1 \in L_1,\ w_2 \in L_2\}$. Конкатенация языка с самим собой называется степенью: $L^k = LL^{k-1}$. Нулевой степенью языка L по определению является язык содержащий только пустое слово. *Итерацией* языка L называется язык L^*, который является объединением всех его степеней: $L^* = \bigcup\limits_{k=0}^{\infty} L^k$.

Недетерминированным конечным автоматом (НКА) называется набор $\mathcal{A} = \langle Q, \Sigma, \delta, F, q_\circ\rangle$, где Q — множество состояний, Σ — входной алфавит, $\delta\colon Q \times \Sigma \to 2^Q$ — функция перехода, $F \subseteq Q$ — множество заключительных состояний, $q_\circ$ — начальное состояние автомата. Последовательность $(q_1, a_1, q_2), \ldots, (q_n, a_n, q_{n+1})$ называется *допустимым путем* в $\mathcal{A}$, если $q_1 = q_\circ$, $q_{i+1} \in \delta(q_i, a_i)$ и $q_{n+1} \in F$. Этому пути соответствует слово $w = a_1 \ldots a_n$. Слово $a_1a_2 \ldots a_n \in \Sigma^*$ *распознается* (допускается) автоматом $\mathcal{A}$, если существует соответствующий этому слову допустимый путь. Каждый автомат *распознает* язык $L(\mathcal{A}) \subseteq \Sigma^*$, который состоит из всех слов, распознаваемых этим автоматом. Язык называется регулярным, если он распознается некоторым автоматом. Множество всех регулярных языков в алфавите Σ обозначается $\mathrm{Reg}(\Sigma)$.

Автоматом с расстояниями над алфавитом Σ называется набор $\mathcal{A} = \langle \Sigma, Q, \rho, q_\circ, F, d\rangle$, где $\langle \Sigma, Q, \rho, q_\circ, F\rangle$ задает недетерминированный автомат, а $d\colon Q \times \Sigma \times Q \to \{0, 1\}$ — функция расстояния. Расстоянием $d(\pi)$ пути $\pi = (q_1, a_1, q_2), \ldots, (q_n, a_n, q_{n+1})$ является сумма расстояний его элементов. Расстоянием $d(w)$ слова $w \in L(\mathcal{A})$ является минимум расстояний по всем допустимым путям, соответствующим слову w. Автомат с расстояниями $\mathcal{A}$ называется *ограниченным*, если существует такая константа M, что $d(w) < M$ для всех слов $w \in L(\mathcal{A})$.

Теорема 1 ([3, 5]). *Свойство ограниченности автоматов с расстояниями алгоритмически разрешимо. Если автомат ограничен и имеет m состояний, то он ограничен константой* $M = 2^{3m^3 + m\lg m + m - 1}$.

В заключении данного раздела приведем определение рационального множества регулярных языков. Пусть заданы два непересекающихся алфавита Σ и Δ. Подстановкой регулярных языков назовем отображение $\varphi\colon \Delta \to \mathrm{Reg}(\Sigma)$, которое сопоставляет каждой букве алфавита Δ некоторый регулярный язык над Σ. Эта подстановка может быть естественным образом расширена до гомоморфизма между свободной полугруппой Δ^+ и конечно порожденной полугруппой регулярных языков $\varphi\colon (\Delta^+, \cdot) \to$

$\to \mathrm{Reg}(\Sigma), \cdot)$. Множество $\mathcal{R}$ регулярных языков над Σ называется *рациональным*, если существует конечный алфавит Δ, регулярный язык $K \subseteq \Delta^+$ и подстановка $\varphi\colon \Delta^+ \to \mathrm{Reg}(\Sigma)$ для которых $\mathcal{R} = \{\varphi(w) \mid w \in K\}$. Рациональные множества регулярных языков являются рациональными подмножествами конечно порожденных полугрупп регулярных языков относительно конкатенации.

2. Задача разложения регулярных языков

Сформулируем задачу разложения регулярных языков по фиксированному базису. Пусть заданы регулярные языки $E_1, \ldots, E_k$. Существует ли алгоритм, позволяющий проверить, возможно ли представить заданный язык L в виде конкатенации языков $\{E_i\}$. Далее мы будем предполагать, что все языки содержат пустое слово. В противном случае длина искомого разложения ограничена сверху длиной самого короткого слова в языке L. Задача разложения связана с такими задачами, как проверка *свойство конечной степени* регулярного языка, проверка ограниченности автомата с расстояниями, разложение языка в произведение простых и «звездных» языков и теоремой Крона—Роудса. Положительное решение было получено в [4].

Теорема 2 ([4]). *Существует алгоритм, который для любого конечного множества $\mathcal{E}$ регулярных языков над алфавитом Σ, любого набора $T \subseteq \{\cdot, \cup, {}^*\}$ языковых операций и регулярного языка L проверяет, возможно ли выразить L через языки $\mathcal{E}$ используя конечное число операций из T.*

Ключевую роль в доказательстве этого утверждения играет разрешимость задачи проверки ограниченности автомата с расстояниями. Алгоритм, который представлен в доказательстве этой теоремы, состоит в построении по языкам L и $\mathcal{E}$ некоторого автомата с расстояниями и проверке всех возможных «коротких» разложений L по базису $\mathcal{E}$, длина которых не превосходит константы, указанной в теореме 1. В [1] алгоритм разложения был расширен для случая рациональных множеств регулярных языков.

Теорема 3 ([1]). *Задача проверки принадлежности регулярного языка L заданному рациональному множеству $\mathcal{R} = \{\varphi(w) \mid w \in K\}$ алгоритмически разрешима.*

Алгоритм проверки принадлежности языка L заданному рациональному множеству, представленный в [1], также основан на проверки свойства ограниченности автомата с расстояниями и имеет двойную экспоненциальную сложность относительно числа состояний недетерминированного автомата, представляющего язык L. Следует отметить, что в работе [6]

была доказана экспоненциальная нижняя оценка сложности для рассматриваемой задачи, и неизвестно эффективных алгоритмов решения данной задачи, даже в случае конечных языков.

3. Возможная криптосхема

В предыдущем разделе было показано, что задача проверки принадлежности языка заданному рациональному подмножеству конечно порожденной полугруппы регулярных языков является вычислительно сложной задачей. В данном разделе приводится описание возможной криптосхемы, построенной на основе этой задачи.

В качестве открытого ключа предлагается взять рациональное множество регулярных языков над алфавитом Σ, заданное множеством образующих и регулярным языком в алфавите образующих Δ. Шифротекстом одного бита является автомат, который представляет 1 или 0 в зависимости от принадлежности рациональному множеству.

Закрытый ключ должен позволять эффективно проверить принадлежность, не обращаясь к проверке свойства ограниченности автомата с расстояниями. Возможным выбором закрытого ключа является конечно порожденная подполугруппа F свободной полугруппы Σ^*. Поскольку такие полугруппы обладают достаточно простой алгебраической структурой, то многие алгоритмические задачи имеют в данном случае эффективное решение. В частности, для любой такой полугруппы можно построить регулярный язык нормальных форм ее элементов и семейство конечных автоматов, так называемую *автоматную структуру*, с помощью которых любое слово может быть приведено к нормальной форме [2]. Это означает, что задача равенства слов для таких полугрупп решается за квадратичное, в зависимости от дины слов, время. Таким образом можно использовать следующую схему генерации ключей:

1) выбрать случайным образом конечное множество $F = \{f_1, \ldots, f_n\}$ слов;
2) построить регулярный язык N нормальных форм полугруппы F^+;
3) выбрать регулярный язык $R \subset N$;
4) построит язык $H = Complement(Fact(F^*))$;
5) случайным образом выбрать регулярные языки $H_i \subseteq H$ для $i = 1, \ldots, n$.

Автоматная структура полугруппы F^+ является закрытым ключом, а языки $H_i \cup \{f_i\}$, R образуют открытый ключ. Для шифрования одного бита требуется построить автомат M, соответствующий слову из R (или

его дополнения R^c). Дешифровка сводится к нахождению самого длинного слова в $F^+ \cap M$ и проверки его принадлежность рациональному подмножеству полугруппы F^+, которое задано языком R. Эта операция может быть эффективно выполнена с использованием автоматной структуры для F^+. Следует отметить, что поскольку эта проверка выполняется системой конечных автоматов, то она может быть эффективно реализована на аппаратном уровне, например, в реконфигурируемых однородных вычислительных структурах.

4. Заключение

В данной работе описывается возможная криптографическая схема с открытым ключом на основе задачи разложения регулярных языков. В данной схеме декодирование имеет полиномиальную сложность, в то время как злоумышленник должен решать задачу двойной экспоненциальной сложности. В тоже время целый ряд вопросов требует дополнительных исследований. Основным вопросом является оценка «реальной» сложности проверки принадлежности языка заданному рациональному подмножеству полугруппы регулярных языков. Также необходимо исследовать зависимость числа состояний автоматов (элементов полугруппы) от длины соответствующего им слова в языке нормальных форм, поскольку число состояний автомата непосредственно связано с коммуникационной сложностью данной схемы.

Литература

[1] *S. Afonin, E. Khazova.* Membership and finiteness problems for rational sets of regular languages // International Journal of Foundations of Computer Science. 2006. V. 17, №. 3. P. 493–506.

[2] *C. M. Campbell, E. F. Robertson, N. Ruškuc, R. M. Thomas.* Automatic semigroups // Theoretical Computer Science. 2001, January. V. 250, №. 1–2. P. 365–391.

[3] *K. Hashiguchi.* Limitedness theorem on finite automata with distance functions // Journal of computer and system sciences. 1982. V. 24. P. 233–244.

[4] *K. Hashiguchi.* Representation theorems on regular languages // Journal of computer and system sciences. 1983. V. 27. P. 101–115.

[5] *H. Leung, V. Podolskiy.* The limitedness problem on distance automata: Hashiguchi's method revisited // Theoretical Computer Science. 2004, January. V. 310, №. 1–3. P. 147–158.

[6] *D. Calvanese, G. De Giacomo, M. Lenzerini, M. Vardi.* Rewriting of regular expressions and regular path queries // Journal of Computer and System Sciences. 2002, May. V. 64. p. 443–465.

Программирование, ориентированное на мониторинг, как элемент контролируемого выполнения аппаратно-программных комплексов

В. А. Галатенко, К. А. Костюхин,
А. С. Малиновский, Н. В. Шмырев

1. Расширение понятия контролируемого выполнения аппаратно-программных комплексов

Под контролируемым выполнением [1] понимается такая организация работы аппаратно-программного комплекса, при которой осуществляется сбор и анализ информации о процессе функционирования и выполняются управляющие воздействия на этот комплекс.

Контролируемое выполнение направлено на выполнение комплексом его миссии, несмотря на наличие ошибок и вредоносных воздействий.

Понятие контролируемого выполнения включает следующие основные положения:

- интеграция средств информационной безопасности, отладки, управления;
- наличие целостного набора средств контролируемого выполнения, возможность взаимодействия между ними;
- охват всех этапов жизненного цикла аппаратно-программных комплексов, включая этап эксплуатации.

В [1] представлена среда контролируемого выполнения (система отладки/мониторинга — COM), которая поддерживает следующие возможности:

- интерактивная отладка;
- мониторинг систем;
- самоконтроль систем;

- детерминированное воспроизведение в рамках многопроцессорной конфигурации;
- применение средств управления информационными системами.

Перечисленные возможности позволяют пользователям и разработчикам аппаратно-программных комплексов получать разнообразную информацию о работе приложений и выполнять, при необходимости, отладочные и управляющие действия. Однако поведение такого рода комплексов описывается большим числом сложных взаимозависимых характеристик, поэтому оценка качества функционирования комплекса в целом, является непростой задачей, даже при наличии детальной информации о различных аспектах его выполнения.

В данной работе представлено развитие концепции контролируемого выполнения, основанное на использовании парадигмы программирования, ориентированного на мониторинг (monitor-oriented programming, MOP [2]).

В последующих разделах данной публикации приводится обоснование данного подхода и представлены основные черты предлагаемого решения.

2. Особенности целевых аппаратно-программных комплексов

Сложность разработки и сопровождения приложений для рассматриваемых в данной работе аппаратно-программных комплексов связана с их особенностями, которые перечислены далее.

1. *Разнородность*. В состав целевых комплексов входят разные аппаратные платформы, имеющие сетевые интерфейсы разных типов и с разной пропускной способностью. Эффективность и предсказуемое поведение целевого комплекса изменяется в зависимости от типа используемых вычислительных платформ и способов их взаимодействия.
2. *Изменяемость*. Здесь можно выделить следующие два аспекта.
   - *Функциональная изменяемость* включает различные варианты реализации одинаковых интерфейсов. Например, узлы обработки сигналов в реальном времени могут по разному обрабатывать входящие данные в зависимости от способности этих узлов обеспечивать надлежащий уровень качества обслуживания (скорость, точность, отказоустойчивость) для входных пакетов определенного размера и реальной пропускной способности сетевого интерфейса.

- *Нефункциональная изменяемость* включает разные конфигурации системных служб. Например, разные политики безопасности, разные политики планирования потоков и процессов узлов реального времени и т. п.

3. *Ограничения, накладываемые требованиями соблюдения определенного уровня качества обслуживания* (*QoS*). Компоненты, осуществляющие контроль выполнения целевых комплексов должны поддерживать требуемый уровень качества обслуживания в автоматическом (без участия оператора) режиме. В частности, эти компоненты должны одновременно обеспечивать соблюдение
   - временных требований для узлов реального времени;
   - требований отказоустойчивости и самовосстановления узлов в результате возникших сбоев;
   - требований информационной безопасности;
   - требований по использованию доступных ресурсов.
4. *Сложность структуры* Аппаратно-программный комплекс может представлять собой надсистему, то есть систему, компонентами которой являются системы. Разработчики отдельных компонентов комплекса далеко не всегда имеют полное представление о нем в целом. В результате могут быть приняты неоптимальные решения, приводящие, например, к повышенному потреблению ресурсов или нарушающие базовые принципы построения системы, например, при использовании узкоспециализированных протоколов.
5. *Динамически изменяемые условия функционирования*. Условия функционирования аппаратно-программных комплексов могут изменяться в ходе выполнения, например, из-за дефицита ресурсов или выявленной угрозы информационной безопасности комплекса. Средства контролируемого выполнения должны обеспечивать своевременное реагирование на подобные ситуации.

С учетом указанных особенностей, при разработке подобного рода аппаратно-программных комплексов целесообразным представляется построение модели, на которой можно проверить выполнение необходимых требований соблюдения определенного уровня качества обслуживания как всего комплекса в целом, так и отдельных его компонентов. В соответствии с этим подходом, в качестве отправной точки контролируемого выполнения предлагается использовать построение формальной модели системы и динамическую верификацию свойств, относящихся к обеспечению надлежащего уровня качества функционирования.

3. Контролируемое выполнение, основанное на моделировании и верификации

Формальные методы верификации программ, основанные на статической проверке соответствия модели, доказательстве теорем, статическом анализе в силу отмеченных в [2] ограничений, не позволяют в полной мере решить задачи, связанные с контролем количественных характеристик выполнения приложений. В данной работе предлагается подход, включающий как статический анализ программного кода на соответствие модели, так и динамическую верификацию программы в ходе ее выполнения. В качестве средства описания модели выбран язык C ACSL (ANSI/ISO C SPECIFICATION LANGUAGE, [3]), который позволяет описывать поведение программ на языке C в виде аннотаций. Инструмент Why [4] вместе со средствами для доказательства утверждений используется для проверки свойств, которые можно проверить статически. Свойства, задающие количественные характеристики выполнения приложения, описываются с использованием средств профилирования инструментального комплекса COM и проверяются динамически в ходе выполнения программы.

Механизм динамической верификации основан на парадигме программирования, ориентированного на мониторинг (MOP). В данной работе механизмы наблюдения и верификации MOP рассматриваются как компоненты среды контролируемого выполнения. Механизм наблюдения включает средства мониторинга и реализованные в рамках данной работы средства профилирования, которые позволяют отслеживать использование приложением целевых ресурсов. Механизм верификации реализован как сопоставление результатов наблюдения с моделью на основе средств самоконтроля.

4. Сбор и анализ информации средствами контролируемого выполнения

Контролируемое выполнение подразумевает наличие средств, осуществляющих постоянный сбор информации о ходе работы целевой системы. К их числу относятся средства самоконтроля программ, средства мониторинга и средства измерения количественных характеристик (профилирование). Для верификации свойств целевой системы могут использоваться данные, собранные при помощи любых из указанных средств.

1. Библиотека средств самоконтроля

Библиотека средств самоконтроля (БСС) реализована в виде набора функций и макровызовов, которые вставляются в исходный текст програм-

мы. Разработчик, исходя из логики поведения программы, может встроить в ее код вызовы БСС для выявления признаков некорректного поведения программы. Для таких ситуаций он может предусмотреть вызов отладочных средств, выдачу диагностики, или, возможно, корректирующие действия для исправления выявленных отклонений от эталонного правильного поведения.

Средства самоконтроля обеспечивают минимальное вмешательство в работу целевой системы, поэтому они широко применяется в приложениях, где важно время выполнения отдельных участков, например, обработчиков прерываний, остановка в которых может сделать дальнейшее функционирование бессмысленным. В отличие от средств мониторинга, средства самоконтроля не требуют внешних программ для взаимодействия с пользователем.

В настоящее время реализован контроль следующих аспектов поведения программ:

- изменение хода выполнения программы в результате некорректных значений внутренних переменных программы и входных параметров вызываемых функций;
- невозможность завершить выполнение фрагмента кода в установленное время (особенно актуально для контроля приложений в системах реального времени);
- сбой в работе программы/системы в результате некорректной работы с динамически выделяемой памятью.

Вызовы БСС разделяются на три основные категории: сенсоры, актуаторы и средства протоколирования.

Сенсором называется пассивный датчик, фиксирующий изменения в системе на внутреннем уровне. Под *актуатором* будем понимать активный датчик, имеющий средства реагирования на происходящие изменения. *Средства протоколирования* доводят до конечного пользователя информацию о состоянии системы и происходящих в ней изменениях.

Ниже рассмотрены основные группы вызовов БСС.

- *Актуаторы*

  Запуск актуаторов осуществляется при помощи вызова *UEL_ACTUATOR*. Использование актуаторов позволяет проверить значение некоторого выражения и, в зависимости от результата, выполнить определенные отладочные действия. Если значение выражения ложно, то вызывается стандартный обработчик или обработчик, заданный пользователем. Стандартный обработчик выполняет следующие действия:

— Помещает в протокол сообщение вида:
ASSERT (*выражение*): [*имя исходного файла*:*номер строки*]

— Генерирует событие типа *INVALID_STATE*, которое может быть отображено средствами мониторинга инструментального комплекса.

— Останавливает выполнение потока/процесса, что позволяет выполнить над ним действия интерактивной отладки. Поскольку исключительной ситуации при этом не возникает, то возможно продолжение выполнения потока в пошаговом или обычном режиме.

- *Сенсоры*
Сенсоры разделяются по приведенным ниже типам.

— *Будильник*

Сенсоры этого типа позволяют проверить, укладывается ли некоторая последовательность вычислений в заданный временной интервал. Сенсор *UEL_SENSOR_GUARD* (*START*) устанавливает интервал времени в секундах. Если за это время управление не достигает вызова *UEL_SENSOR_GUARD* (*STOP*), то в протокол помещается соответствующее сообщение и генерируется событие, отображаемое средствами мониторинга.

— *Профилирование*

Сенсоры этого типа отмечают начало (*UEL_SENSOR_PROFILER* (*START*)) и конец (*UEL_SENSOR_PROFILER* (*STOP*)) профилируемого участка кода. Время выполнения участка (вместе с текстовым описателем) выводится в протокол.

— *Контроль использования динамически распределяемой памяти*

Функции запроса и освобождения памяти *UEL_calloc*, *UEL_malloc*, *UEL_realloc* и *UEL_free* аналогичны стандартным, но регистрируют проходящие через них объекты, сохраняя их адреса, размеры и позицию исходного кода, где они были вызваны. Они также порождают событие *INVALID_STATE* и выдают в протокол сообщения в случае некорректных ситуаций, например, если функции *UEL_free* или *UEL_realloc* передан указатель на свободную память.

Сенсор *UEL_SENSOR_CHECK_MEMORY_INTEGRITY* позволяет запомнить текущее состояние памяти, а позже вывести в протокол отчет об изменениях, произошедших с момента последнего сохранения.

— *Динамическая проверка указателей*

Сенсоры типа *UEL_SENSOR_CHECK_MEMORY_ACCESS* служат для проверки корректности указателей. Их целесообразно использовать, например, для проверки аргументов функции, в сочетании с актуатором.

Возвращаемые значения — UEL_ERR (ошибка при проверке), UEL_BAD (некорректный указатель), UEL_GOOD (корректный указатель).

При помощи вызовов данной группы можно проверить, что указатель содержит адрес корректной для потока области памяти,

* содержащей программный код;
* из которой можно прочесть заданное число байтов;
* в которую можно записать заданное число байтов;
* в которой находится строка печатных символов заданной длины (или до символа конца строки).

- *Средства протоколирования*

  В любой точке программы можно вывести отчет о текущем состоянии динамически распределяемой памяти с помощью вызова *UEL_LOG_MEMORY*.

  Чтобы поместить интересующую пользователя информацию в протокол, нужно воспользоваться вызовом *UEL_LOG_EXPRESSION*.

  Еще один вызов *UEL_LOG_DEBUG_ONLY* служит для вычисления заданного выражения только при включенной отладке. Таким способом удобно задавать, например, отладочную печать.

- *Прочее*

  UEL_init, *UEL_fini* — инициализация и терминирование БСС.

Такой набор функций БСС был выбран авторами на основе опыта, полученного при отладке распределенных приложений, некоторые компоненты которых работают под управлением системы реального времени. В частности, необходимо иметь средства контроля времени выполнения определенных фрагментов кода, причем как пассивные (сенсоры), так и активные (актуаторы), реагирующие на соответствующие данные сенсоров. Контроль изменения и доступа к памяти необходим, если отлаживается приложение, активно использующее память, то есть, практически любое довольно большое приложение. Механизмы, предоставляющие различные вспомогательные средства (например, средства протоколирования) удобно использовать при управлении отладкой (их можно одновременно включить или выключить одним действием).

2. Система профилирования

В рамках инструментального комплекса COM создана система профилирования целевых систем, функционирующих в условиях дефицита ресурсов. Возможности данной системы включают в себя

1. Поддержку единого способа хранения информации, получаемой из разных источников (система профилирования должна интегрироваться в инструментальный комплекс COM, работающий с разными источниками информации: агент мониторинга, агент профилирования и подобные им).
2. Сбор следующих количественных характеристик:
   - процессорное время, затрачиваемое на выполнение определенных фрагментов кода программы;
   - память, используемая программой;
   - аппаратные события (при их наличии), возникающие в ходе выполнения программы;
   - сетевые интерфейсы, используемые программой.
3. Использование аппаратных возможностей профилирования при условии наличия соответствующий поддержки.
4. Использование стандартных средств и механизмов для возможности последующей интеграции с существующими решениями.

2.1. Архитектура системы профилирования

Система профилирования представляет собой распределенное приложение, состоящее из:

- компонента ядра профилируемой системы (драйвера);
- агента профилирования, выполняющегося на целевой машине (демона профилирования);
- библиотеки профилирования, содержащей вызовы функций для инструментовки кода;
- менеджера профилирования на инструментальной стороне;
- базы данных для хранения собранной информации (БД событий);
- программ-анализаторов собранной информации, представляющих полученную информацию как в графическом, так и в текстовом виде.

Архитектура системы профилирования представлена на рисунке 1.

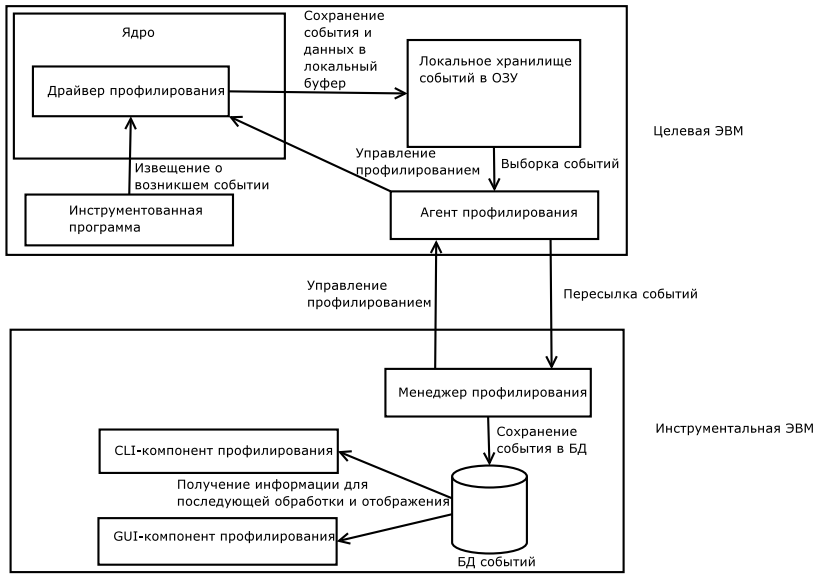


Рис. 1. Архитектура системы профилирования

2.2. События профилирования

Различают простые и составные события профилирования. Простое событие содержит информацию ровно об одном атомарном событии профилирования (атомарными событиями являются: аппаратное событие профилирования, отправка/получение одного сетевого пакета, одна операция выделения/освобождения памяти). Составное событие состоит более чем из одного атомарного события, например, «с помощью данного сетевого интерфейса получено 52 пакета общим размером 3012 байт» или «произошло 103034 аппаратных событий выполнения одной команды процессора». Составные события порождаются при помощи вызовов библиотеки профилирования, а также с использованием аппаратных возможностей целевой платформы. В частности события, состоящие из аппаратных событий профилирования, порождаются при помощи специальных аппаратных счетчиков событий, порождающих прерывание при переполнении.

2.3. Физические компоненты системы и их взаимодействие

На целевой системе в ядро встроен *драйвер профилирования*, осуществляющий предварительные действия по профилированию (создать/удалить/обнулить локальное хранилище событий), управление профили-

рованием (начать/остановить), первичную обработку и сохранение возникающих в пользовательской программе событий профилирования. Он также осуществляет обработку прерываний и исключительных ситуаций, связанных с профилированием. Организация локального хранилища непосредственно зависит от типа событий профилирования, которые желает протоколировать пользователь. Так для протоколирования составных событий, состоящих из аппаратных событий, создается массив адресов, каждой ячейке которого соответствует адрес в сегменте кода целевой системы. Если событие профилирования возникает по адресу, находящемуся в массиве, то соответствующий элемент массива инкрементируется. В случае, если событие профилирования возникает по адресу, не находящемуся в массиве, инкрементируется специально выделенный (как правило, последний) элемент массива. Для событий профилирования, связанных с использованием памяти и сетевых интерфейсов, локальное хранилище создается минимального размера и состоит из записей, содержащих идентификатор события (см. ниже) и текущего количества байтов, относящихся к данному событию. Для событий профилирования памяти это разница между выделенным и освобожденным числом байтов, а для сетевого интерфейса — это постоянно инкрементируемое число переданных или полученных байтов.

С драйвером взаимодействует *агент профилирования*, передающий управляющие запросы от менеджера профилирования. Взаимодействие основано на установке специальных системных переменных, значения которых отслеживает драйвер. Кроме того, агент профилирования читает информацию из локального хранилища и передает ее менеджеру. Для работы с драйвером агент применяет вызовы библиотеки профилирования, которая может использоваться непосредственно в пользовательской программе.

На инструментальной ЭВМ управление сбором событий профилирования осуществляет *менеджер профилирования*. Менеджер профилирования является стандартным компонентом комплекса COM. Он взаимодействует с остальными компонентами комплекса (таблицей имен, компонентами сред выполнения). Менеджер профилирования передает агенту пользовательские запросы на запуск/останов/сброс профилирования, а также на изменение маски событий профилирования. Также менеджер профилирования сохраняет полученную информацию в БД событий.

БД событий хранит информацию обо всех событиях целевых систем (включая события мониторинга) в едином формате.

Внешние компоненты профилирования. Открытость интерфейсов предполагает использование программ сторонних разработчиков, использующих те же самые интерфейсы.

В настоящее время вместе с системой профилирования используется набор утилит oprofile, позволяющий в режиме командной строки управлять счетчиками аппаратных событий целевой системы, а также осуществлять сбор событий профилирования, содержащих информацию об использовании динамически выделяемой памяти и сетевых интерфейсов целевой системы.

Данный набор утилит был дополнен утилитой удаленного управления профилированием ropcontrol (аналогом штатной утилиты opcontrol). Был сохранен пользовательский интерфейс opcontrol и реализованы дополнительные опции командной строки, позволяющие осуществлять взаимодействие с агентом профилирования на целевой стороне.

Итогом использования oprofile является файл в формате gmon.out, который далее может быть обработан стандартной утилитой gprof. Пример выполнения этой утилиты приведен ниже.

```
 1 Flat profile:
 2
 3 Each sample counts as 1 samples.
 4  %   cumulative  self          self    total
 5 time  samples  samples  calls T1/call T1/call name
 6 28.22   92.00   92.00                          nfs_sndlock
 7 17.79  150.00   58.00                          copyto
 8 14.11  196.00   46.00                          _assert
 9 13.80  241.00   45.00                          fcntl
10 10.73  276.00   35.00                          timer_settime
11  4.60  291.00   15.00                          copyfrom
12  4.60  306.00   15.00                          elfload
13  3.37  317.00   11.00                          mtxLockInternal
14  2.78  326.00    9.00                          mqSend
```

В приведенном примере выборка производилась через каждые 750000 аппаратных событий выполнения команд процессора.

В силу специфики рассматриваемого класса целевых систем, понятие профилирования в рамках данной работы используется в более широком смысле, чем сбор информации только об использовании процессорного времени. Под профилированием понимается сбор любых количественных характеристик выполнения приложения, связанных с расходованием ресурсов вычислительной среды, таких как память, полоса пропускания сети и др.

Результаты профилирования служат входными данными для средств верификации, осуществляющих сравнение фактического поведения при-

ложения с эталонным. Помимо этого они могут использоваться традиционным способом, т.е. для построения профилей с целью оптимизации приложений, что особенно важно, если целевой аппаратно-программный комплекс содержит узлы, работающие в реальном масштабе времени в условиях дефицита ресурсов.

С этой точки зрения важно, чтобы результат работы средств профилирования был представлен в унифицированном формате, позволяющем использовать не только средства инструментального комплекса, осуществляющего контролируемое выполнение целевой системы, но и внешние инструменты обработки данных профилирования.

Необходимо также использовать стандартизованные (в том числе аппаратные) интерфейсы для получения информации о ходе выполнения программы, а именно: интерфейсы доступа к аппаратным счетчикам событий процессора, стандартные библиотеки компилятора и целевой ОС, обеспечивающие поддержку построения профилей выполнения программы.

Применение стандартизованных подходов, существующих в различных предметных областях, является одной из важных характеристик среды контролируемого выполнения, поскольку это повышает переносимость ее инструментальных средств и позволяет наращивать функциональность за счет применения средств сторонних разработчиков.

5. Выводы

В представленной работе понятие контролируемого выполнения расширено за счет внедрения парадигмы программирования, ориентированного на мониторинг, введения средств динамического моделирования целевых аппаратно-программных комплексов, а также новых средств профилирования. Под профилированием в контексте данной работы понимается сбор и анализ информации о различных количественных характеристиках выполнения комплексов, включая данные об использовании как процессорного времени, так и других целевых ресурсов.

Необходимость введения этих средств продиктована как сложностью рассматриваемых аппаратно-программных комплексов, так и критичностью предъявляемых к ним требованиями по обеспечению должного качества обслуживания.

Литература

[1] *Костюхин К. А.* Организация контролируемого выполнения для разнородных распределенных программно-аппаратных комплексов // Ph.D. thesis. Научно-исследовательский институт системных исследований, 2006.

[2] *Chen F., Rosu G.* MOP: Reliable software development using abstract aspects // Tech. rep. Dept. of Computer Science, University of Illinois, 2006.

[3] *Baudin P., Filliatre J.-C., Marche C. et al.* ASCL: ANSI/ISO C Specification Language, 2008.

[4] *Filliatre J.-C., Marche C.* The why/krakatoa/caduceus platform for deductive program verification. OOPSLA, 2004.

Модель управления рисками информационной безопасности на основе знания угроз

А. В. Львова

1. Введение

Задача управления рисками неблагоприятных событий является одной из важнейших при проведении аудита безопасности информационных активов организации, заказываеющей такой аудит. Основная задача аудита — объективно оценить текущее состояние информационной безопасности (ИБ), а также ее адекватность задачам, которые она рпизвана решать. В этой связи под аудитом ИБ понимается системный процесс получения количественных и качественных оценок в соответствии с определенными критериями [3]. В области ИБ большое распространение получило использование показателей рисков неблагоприятных событий (далее для краткости именуемых рисками) для угроз ИБ. Значение риска, являющееся произведением стоимости защищаемого ресурса на вероятность реализации угрозы в его адрес, служит показателем полноты, комплексности и эффективности системы ИБ организации. Оно может свидетельствовать о текущем уровне защиты, позволяет выявить ее слабые места.

Существуют различные способы проведения оценки рисков. Они отличаются методами оценивания их составляющих — стоимости и вероятности. Наиболее распространено использование экспертных оценок в совокупности с балльными шкалами значений [3], что затрудняет трактовку результатов расчетов. Эффективность анализа рисков снижает также рассмотрение типовых угроз ИБ применительно к конкретной организации с характерными для нее информационными ресурсами. В связи с этим обстоятельством автором предлагается модель проведения анализа рисков, основывающаяся на рассмотрении только реальных угроз ИБ и ее стоимостных оценок в денежном выражении.

Первоначально необходимо выявить противников, которые реально действуют в окружающей среде организации и заинтересованы (или уже делали попытки) в нарушении конфиденциальности, целостности или доступности ее информационных ресурсов. На этом основании выделяются

ресурсы, требующие защиты. Как правило, они изначально являются наиболее значимыми. Далее рассматриваются варианты получения доступа противника к ресурсам и способы их дальнейшего использования, противоречащие интересам организации. Выполнение действий по получению доступа и успешному использованию ресурса дает способ реализации реальной угрозы противником. Далее по каждой угрозе эксперты оценивают стоимость ресурса в денежном выражении. Вероятности реализации угроз предлагается рассчитывать на основании статистики инцидентов, собранной внутри данной организации, экспертных оценок стоимости и качества используемых средств защиты, а также анализа мотивации противника (рассчитанной на получение финансовой выгоды и психологической) [1]. В результате оценки принимают вид реальных стоимостных показателей существующих угроз.

Модель анализа рисков может использоваться для управления рисками ИБ организации. Управление заключается в контроле за уровнями рисков по выявленным угрозам и принятии мер по своевременному реагированию на изменение ситуации. Для этого необходимо поддерживать актуальность исходных данных модели, накапливать статистическую информацию об инцидентах в области ИБ организации, проводить разведывательную работу по выявлению угроз. Такой подход позволяет контролировать текущую ситуацию по защищенности ресурсов, быстро реагировать на изменения угроз, а также планировать перераспределение затрат на защиту информацинных активов в пользу снижения уровня рисков и повышения экономической эффективности системы обеспечения ИБ.

2. Основные принципы

1. Модель управления рисками оперирует конкретными злоумышленниками и конкретными угрозами, исходящими от них.
2. Все показатели (ущерб, выгода и подобные им) должны быть количественно оцениваемы в денежном эквиваленте с приемлемой точностью.
3. Количественные оценки вероятностей реализации угроз основываются на анализе мотивации противника и обобщении собственной статистики инцидентов в подконтрольной организации в области ИБ.
4. Степень важности и актуальности конкретной угрозы определяется только величиной финансовых потерь в случае ее реализации.
5. В качестве показателя экономической эффективности вложений в ИБ предлагается использовать рентабельность как отношение

прибыли от функционирования системы обеспечения ИБ к общим затратам на ее создание и эксплуатацию.

6. Критерием достижения цели управления рисками может служить максимальная рентабельность, минимальные суммарные риски при ограниченных затратах на ИБ, минимальные затраты на ИБ при фиксированных рисках.

3. Алгоритм управления рисками

При описании алгоритма используются следующие условные обозначения:

t (trespasser) — конкретный противник организации, мотивированный на получение выгоды от реализации угрозы деструктивного воздействия на определенную информацию;

i (information) — информация, интересующая противника;

r (resource) — ресурс организации (физический, технический, персонал, в том числе носители информации);

$d(i)$, $d(r)$ (doing) — действие противника по отношению к информации или ресурсу;

$s(i_1, \ldots, i_n)$, $s(r_1, \ldots, r_n)$ (security facility) — средство (мера) защиты (СЗ) по отношению к информации или к ресурсам;

$n(s)$ — количество ресурсов, защищаемых СЗ;

$M(i, t)$ (method) — способ реализации угрозы (РУ) противником в отношении информации. Включает определенный способ доступа и способ использования информации (ИИ): $M(i, t) = (Ma(i, t); Mr(i, t))$;

$Ma(i, t)$ (access) — способ доступа противника к информации; доступ осуществляется через ресурсы r с помощью действий $d(r)$;

$Mr(i, t)$ (realization) — способ использования информации противником; ИИ производится с помощью действия d(i);

$o(M)$ (occurrence) — инцидент в области ИБ: $o(M) = (o(Ma); o(Mr))$;

$N(M) = N(o(M))$ (number) — количество инцидентов в области ИБ с использованием способа РУ, зафиксированных в статистических данных;

$N(Ma)$, $N(Mr)$ — количество инцидентов в области ИБ с использованием способа доступа и способа ИИ:

$N(Ma) = Ns(Ma) + Nf(Ma)$, $N(Mr) = Ns(Mr) + Nf(Mr)$, где

$Ns(Ma)$, $Ns(Mr)$ (success) — количество успешных атак в области ИБ с использонием способа доступа и способа ИИ;

$Nf(Ma)$, $Nf(Mr)$ (failure) — количество предотвращенных атак в области ИБ с использованием способа доступа и способа ИИ;

$G(M)$ (gain) — экспертная оценка выгоды, получаемой противником от РУ;

$V(M)$ (value) — оценка стоимости для противника РУ: $V(M) = V(Ma) + V(Mr)$;

$V(Ma)$ — оценка стоимости для противника получения доступа способом Ma;

$V(Mr)$ — оценка стоимости для противника использования информации способом Mr;

Для расчета оценок стоимости используются следующие экспертные оценки:

$Ve(s(i))$, $Ve(s(r))$ — экспертные оценки стоимости «взлома» средства защиты s на ресурсе r и для информации i;

$Ve(r, d(r))$ — экспертная оценка затрат противника на осуществление действия d на ресурс r, не связанных со «взломом» средств защиты.

$Ve(i, d(i))$ — экспертная оценка затрат противника на осуществление действия d над информацией i после получения доступа.

$O(i, t)$ (opposition) — экспертная оценка возможных финансовых потерь противником t, получившим и использующим информацию i, вследствие контрдействий владельца информации, применения им компрометирующей информации, данных службы безопасности, полученных методами агентурной и технической разведки;

$P(M)$ (probability) — оценка вероятности РУ способом . Оценка вероятности РУ рассчитывается на основании статистики, инцидентов, анализа мотивации противника и психологической предрасположенности противника;

$P^1(M)$ — компонента вероятности, рассчитанная на основании статистики инцидентов, рассчитывается как вероятность выполнения двух совместных событий: получения доступа и использование информации;

$P^2(M)$ — компонента вероятности рассчитанная на основании анализа мотивации противника; а именно оценок выгоды от РУ, затрат на РУ, потерь от контрдействий;

$P^3(M)$ — компонента вероятности, оцениваемая экспертами и показывающая психологическую предрасположенность противника на РУ;

$L(M)$ (loss) — оценка финансовых потерь владельца информации от РУ способом M: $L(M) = L(M) + L(Mr)$, где:

$L(Ma)$ — оценка финансовых потерь за счет нарушения средств защиты, ресурсов в процессе получения доступа;

$L(Mr)$ — оценка финансовых потерь за счет использования информации противником.

Для расчета оценок финансовых потерь используются следующие статистические данные:

$L(o(Ma))$ (occurrence loss) — ущерб владельца от нарушения ресурсов в процессе успешного доступа в инциденте $o(M)$;

$L(o(Mr))$ — ущерб владельца от успешной РИ в инциденте $o(M)$ (судебные издержки, командировки, смена персонала и т.д.).

Для расчета оценок финансовых потерь организации от деструктивных действий противника используются следующие экспертные оценки:

$Le(r, d(r))$ (expert) — экспертная оценка потерь от действия d на ресурс r;

$Le(i, d(i))$ — экспертная оценка финансовых потерь за счет использования информации противником.

R (risk) — риск; $\widetilde{R}$ — нериск;

$R(M)$ — оценка риска РУ способом M, является произведением вероятности РУ на ущерб от РУ;

$R(i)$ — оценка риска для информации i, рассчитывается как максимальный риск РУ для данной информации;

$\widetilde{R}(M)$ — оценка нериска РУ способом M, является произведением вероятности не реализации угрозы на ущерб от РУ;

$\widetilde{R}(i)$ — оценка нериска для информации i, рассчитывается как минимальный нериск РУ для данной информации.

C (cost) — оценка суммарной стоимости средств (мер) защиты;

$C(s)$ — оценка финансовых затрат на средство защиты s; включает в себя:

$Ci(s)$ (install) — экспертная оценка стоимости покупки, установки и настройки СЗ (внедрения меры защиты) на один ресурс;

$Cu(s)$ (use) — экспертная оценка стоимости эксплуатации СЗ в год;

$Cr(s)$ (removal) — экспертная оценка стоимости вывода из эксплуатации СЗ;

$I(s)$ (indirect gain) — экспертная оценка косвенной выгоды от внедрения СЗ (сокращение штата, ПО и другие);

Profit (profitability) — оценка рентабельности системы ИБ организации. Рентабельность системы ИБ организации представляется собой отношение прибыли от использования СЗ (мер защиты) к общим затратам на ее создание и эксплуатацию. В качестве прибыли рассматривается разность между выгодой и затратами. Автором предлалается оценивать выгоду суммарными нерисками по информации, подвергающейся угрозам противников, в сумме с косвенной выгодой от внедрения СЗ.

Алгоритм управления рисками выглядит следующим образом.

1. Выявление реальных противников t, $t \in \{t_1, \ldots, t_{nt}\}$. Характеристика противников.
2. Описание структуры организации:

а) описание значимых ресурсов r, $r \in \{r_1, \ldots, r_{nr}\}$;

б) описание информации i, $i \in \{i_1, \ldots, i_{ni}\}$;

в) описание действий противника над ресурсами $d(r)$, $d(r) \in \in \{d_{1r}, \ldots, d_{ndr}\}$ и над информацией $d(i)$, $d(i) \in \{d_{1i}, \ldots, d_{ndi}\}$;

г) описание используемых средств защиты $s(i_a, \ldots, i_b)$, $s(r_a, \ldots, r_b)$, $s \in \{s_1, \ldots, s_{ns}\}$ для информации и ресурсов.

3. Получение экспертных оценок:

а) $Ve(r, d(r))$, $Ve(i, d(i))$ — затраты противника на осуществление действий над ресурсами и над информацией, не связанных со «взломом» СЗ, $r \in \{r_1, \ldots, r_{nr}\}$, $d(r) \in \{d_{1r}, \ldots, d_{ndr}\}$, $i \in \in \{i_1, \ldots, i_{ni}\}$, $d(i) \in \{d_{1i}, \ldots, d_{ndi}\}$;

б) $V(s(r))$, $V(s(i))$ — затраты противника на «взлом» СЗ ресурса и информации соответственно; $s \in \{s_1, \ldots, s_{ns}\}$;

в) $Ci(s)$, $Cu(s)$, $Cr(s)$ — затраты владельца информации на СЗ, $s \in \in \{s_1, \ldots, s_{ns}\}$;

г) $I(s)$ — косвенная выгода от внедрения СЗ, $s \in \{s_1, \ldots, s_{ns}\}$;

д) $Le(r, d(r))$, $Le(i, d(i))$ — ущерб владельца от деструктивных действий противника по отношению к ресурсами и информации, $r \in \{r_1, \ldots, r_{nr}\}$, $d(r) \in \{d_{1r}, \ldots, d_{ndr}\}$, $i \in \{i_1, \ldots, i_{ni}\}$, $d(i) \in \in \{d_{1i}, \ldots, d_{ndi}\}$.

4. Описание способов РУ.

а) Выявление способов доступа. Доступ реализуется через некоторые ресурсы с помощью определенных действий:

$$M(i, t) = \{(r; d(r) | r \in \{r_1, \ldots, r_{nr}\}, \quad d(r) \in \{d_{1r}, \ldots, d_{ndr}\}\}).$$

б) Выявление способов использования информации:

$$Mr(i, t) = (i; d(i)), \quad i \in \{i_1, \ldots, i_{ni}\}, \quad d(i) \in \{d_{1i}, \ldots, d_{ndi}\}.$$

в) Определение способов реализации угроз как комбинации способов доступа и способов использования информации:

$$M(i, t) \in \{M_1, \ldots, M_{nm}\}; \quad M(i, t) = \{Ma(i, t);$$
$$Mr(i, t) | Ma(i, t) \in \{Ma_1, \ldots, Ma_{nma}\}, \ Mr(i, t) \in \{Mr_1, \ldots, Mr_{nmr}\}\}.$$

г) Экспертная оценка параметров, характеризующих способы РУ: $M(i, t) \in \{M_1, \ldots, M_{nm}\}$: $G(M)$, $P3(M)$, $O(i, t)$, где $P^3(M) \in [0; 1]$, и по умолчанию $O(i, t) = 0$.

5. Фиксация значений:

   а) фиксируется конкретная информация i, $i \in \{i_1, \ldots, i_{ni}\}$;

   б) фиксируется конкретный противник t, $t \in \{t_1, \ldots, t_{nt}\}$;

   в) фиксируется конкретный способ РУ, а именно — $M(i, t) = \{Ma; Mr\}$, $M \in \{M_1, \ldots, M_{nm}\}$.

6. Анализ статистических данных об инцидентах в области ИБ.

   а) Описание инцидентов в области ИБ: $o(M)$, $o(M) \in \{o_1, \ldots, o_{no}\}$, $o(M) = (o(Ma); o(Mr))$.

   б) Указание ущерба владельца по каждому инциденту: $L(o(Ma))$, $L(o(Mr))$.

   в) Подсчет количества успешных и предотвращенных атак по инцидентам: $Ns(Ma)$, $Nf(Ma)$, $Ns(Mr)$, $Nf(Mr)$.

   Дальнейшие расчеты производится при наличии достаточного объема статистических данных:

$$Ns(Ma) + Nf(Ma) = N_a > 0,$$
$$Ns(Mr) + Nf(Mr) = N_r > 0,$$

   по умолчанию $Na = Nr = 1$.

   г) Расчет оценки компоненты вероятности РУ на основании статистики инцидентов:

$$P^1(M) = P^1(Ma) \cdot P^1(Mr),$$
$$P^1(Ma) = \frac{Ns(Ma)}{Ns(Ma) + Nf(Ma)},$$
$$P^1(Mr) = \frac{Ns(Mr)}{Ns(Mr) + Nf(Mr)}.$$

7. Анализ мотивации противника на реализацию угрозы.

   а) Расчет стоимости РУ для противника:

$$V(M) = V(Ma) + V(Mr),$$

   где $V(Ma)$ — оценка стоимости получения доступа, вычисляется как сумма экспертных оценок реализации деструктивных действий и «взлома» СЗ ресурсов, входящих в данный способ доступа, а также информации, к которой осуществляется доступ:

$$V(Ma) = \sum_r Ve(r, d(r)) + \sum_r sum_s Ve(s(r)) + Ve(s(i));$$

   $V(Mr)$ — оценка стоимости РИ, равная экспертной оценке выполнения действия над информацией после получения доступа:

$$V(Mr) = Ve(i, r(i)).$$

б) Расчет оценки компоненты вероятности РУ на основании анализа мотивации противника:

$$P^2(M) = \frac{\sum_v k_v \cdot P^v(M)}{\sum_v k_v},\ v = 1 \ldots 3.$$

Коэффициенты:

k_1 — статистический;

k_2 — мотивационный;

k_3 — психологический.

Коэффициенты рассчитываются следующим образом:

$k_v = 0$, если P^v не рассчитано, $v = 1, 2, 3$;

$k_1 = \log_N (N_a + N_r)$ при $N_a + N_r \leqslant N$;

$k_1 = 1$ при $N_a + N_r > N$;

$k_v = 1$, если P^v рассчитано, $v = 2, 3$.

N — граничное значение, задается экспертом, по умолчанию $N = 10$.

8. Расчет оценок риска и нериска для способа РУ.

а) Расчет оценки ущерба от РУ для владельца информации:

$$L(M) = L(Ma) + L(Mr);$$

$L(Ma)$ — ущерб владельца от получения доступа, рассчитывается на основании статистики инцидентов (усредненные по статистике успешных инцидентов финансовые потери от получения доступа способом Ma) и на основании суммарных потерь от действий d на ресурсы r, через которые осуществляется доступ;

$$L(Ma) = \frac{\frac{k_1}{Ns(Ma)} \cdot \sum_o L(o(M(o))) + \sum_r Le(r,\ d(r))}{k_1 + 1},\ r \in Ma;$$

$L(Mr)$ — ущерб владельца от ИИ, рассчитывается на основании статистики инцидентов (усредненные по статистике успешных инцидентов финансовые потери от ИИ способом Mr) и на основании экспертной оценки потерь за счет действия по использованию информации;

$$L(Mr) = \frac{\frac{k_1}{Ns(Mr)} \cdot \sum_o L(o(M(r))) + Le(i,\ d(i))}{k_1 + 1},\ r \in Mr;$$

где k_1 — статистический коэффициент.

б) Расчет оценок риска и нериска для способа РУ:

$$R(M) = L(M) \cdot P(M),\ \widetilde{R}(M) = L(M) \cdot [1 - P(M)].$$

9. Выбор следующих не рассмотренных значений.

   а) Переход к п. 4.3 и выбор следующего способа РУ $M(i,\ t)$. В случае рассмотрения всех значений переход к п. 9.2.

   б) Переход к п. 4.2 и выбор следующего противника t. В случае рассмотрения всех значений переход к п. 10.

10. Нахождение риска и нериска для информации:

$$R(i) = \max_M R(M(i,\ t)) = R(M^{\max}(i));$$

$$\widetilde{R}(i) = \min_M \widetilde{R}(M(i,\ t)) = \widetilde{R}(M^{\max}(i)),\ M \in \{M_1,\ \ldots,\ M_{nm}\}.$$

11. Переход к п. 5.1 и выбор не рассмотренной информации i. В случае рассмотрения всех значений переход к п. 12.

12. Нахождение максимального и минимального риска для системы ИБ организации:

$$R^{\max} = \max_i R(i),\ \text{ что соответствует способу РУ } M^{\max};$$

$$R^{\min} = \min_i R(i),\ \text{ что соответствует способу РУ } M^{\min},$$

$$i \in \{i_1,\ \ldots,\ i_{ni}\}.$$

13. Вычисление оценки рентабельности системы ИБ:

$$\text{Profit} = \frac{K \cdot \sum_i \widetilde{R}(i) + \sum_s I(s) - C}{C},\ \text{ где}$$

$$C = \sum_s C(s) = \sum_s n(s) \cdot (Ci(s) + Cu(s)),$$

$$s \in \{s_1,\ \ldots,\ s_{ns}\},\quad i \in \{i_1,\ \ldots,\ i_{ni}\}.$$

K — нормировочный коэффициент.

Коэффициент K вводится для приведения значения суммарных нерисков в рамки максимально возможных потерь. Так как в рассмотрении участвует различная информация и угрозы, среди возможных событий могут встречаться несовместные. Общее значение нериска при суммировании может превысить максимальные потери (стоимость бизнеса). Предлагается рассчитывать коэффициент следующим образом:

$$K = \frac{\max_i L(M^{\max}(i))}{\sum_i L(M_{\max}(i))}.$$

14. Анализ результатов.

а) Если получено, что $\text{Profit} \leqslant 0$, затраты на систему защиты превышают выгоду от ее функционирования. Этот факт может инициировать уменьшение текущих затрат на поддержание ИБ организации. Выбирается СЗ, соответствующее способу РУ с минимальным риском $R^{\min}$, и моделируется процесс фукнционирования системы обеспечения ИБ организации без этого средства или снижение затрат на него (эксплуатационных расходов). Процедура повторяется с п. 4 до получения положительной рентабельности.

б) Если получено, что $\text{Profit} > 0$, можно попытаться повысить экономическую эффективность системы защиты. Для этого рассматривается способ РУ с максимальным риском $R^{\max}$ и моделируется введение нового СЗ на его предотвращение. Процедура повторяется с п. 4.

Если значение рентабельности при этом увеличилось — затрата на использование этого средства принимается, если уменьшилось — отклоняется. Процедура повторяется до тех пор, пока рентабельность не достигнет некоторого порогового значения.

В качестве дополнительного критерия может быть использовано значение максимального риска или суммарных рисков. Наличие дополнительного критерия по такому важному показателю безопасности, как оценка риска, позволяет контролировать его уровень при работе над повышением экономической эффективности. По дополнительным критериям могут быть заданы пороговые значения. Дополнительный критерий может также выступать и в роли главного.

Оценка параметров в п. 2.2, 3.4 алгоритма проводится экспертом (группой экспертов) в денежных единицах с приемлемой точностью. Для объединения оценок нескольких экспертов по одному параметру предлагается использовать метод групповой оценки [2].

4. Апробация алгоритма

Апробация предложенной модели управления рисками ИБ организации была проведена совместно с Информационно-Вычислительным Центром МЭИ (ТУ) в рамках корпоративной компьютерной сети ИВС МЭИ (ТУ). Для проведения моделирования и расчетов параметров была разработана инструментальная система «Аудит». Система предназначена для использования специалистами и экспертами по ИБ.

В результате анализа множества угроз, актуальных для ИВС, и руководствуясь мнением экспертов, в качестве реальных угроз, пригодных для

числового анализа, были выбраны спам и вирусы. Исходными данными для проведения анализа послужили статистические данные обнаружения вирусов и спама в общеуниверситетской системе электронной почты, обнаружение вирусов антивирусными серверами, а также обнаружение спама и вирусов пользователями вручную. Использовались экспертные оценки в денежном выражении, такие как стоимость средств защиты информации, ущерб от реализации угроз, вероятность срабатывания защиты.

Для оценки качества системы ИБ ИВС МЭИ (ТУ) были рассмотрены все возможные комбинации используемых средств защиты от реальных противников (см. таблицу).

СЗ	Угрозы / Риск	Спам с пометкой		Спам без пометки		Вирус		Стоимость СЗ	Рентабельность
		Нериск	Риск	Нериск	Риск	Нериск			
1	Spam-Assassin Sophos, Symantec	51,1	782,2	47,8	785,5	1,35	51,7	49,1	31,9
2	Spam-Assassin Sophos	51,1	782,2	47,8	785,5	5,57	47,4	38,2	41,2
3	Spam-Assassin Symantec	51,1	782,2	47,8	785,5	23,6	29,4	12,5	127,0
4	Spam-Assassin	51,1	782,2	47,8	785,5	28,1	24,9	1,70	914,0
5	Sophos, Symantec	0	833,3	833,3	0	1,35	51,7	47,3	17,7
6	Sophos	0	833,3	833,3	0	5,57	47,4	36,5	23,1
7	Symantec	0	833,3	833,3	0	23,6	29,4	10,7	79,1

В качестве показателя экономической эффективности при анализе использовался критерий рентабельности. Анализ показал, что максимальное значение рентабельности достигается при использовании только некоммерческих средств защиты. Причина в их низкой стоимости эксплуатации (вариант 4). При этом риски по другим направлениям оказываются недопустимо большими. Минимальная рентабельность соответствует использованию только коммерческих средств (вариант 5). При этом по другим направлениям возникают недопустимые риски. Для исходной конфигурации системы защиты, используемой в ИВС (вариант 1), характерно использование как коммерческих, так и свободно распространяемых продуктов, что является наилучшим вариантом.

Таким образом, проверка модели на реальной корпоративной системе и реальных статистических и экспертных данных показала, что предлагаемые к использованию показатели являются оцениваемыми, адекватными

поставленной задачи, дающими реальный и полезный практический результат.

Литература

[1] *Бородюк В. П.*, *Львова А. В.* Повышение экономической эффективности системы информационной безопасности // Вестник МЭИ. 2007. № 4.

[2] *Евланов Л. Г.*, *Кутузов В. А.* Экспертные оценки в управлении. М.: Экономика, 1978.

[3] *Петренко С. А.*, *Симонов С. В.* Управление информационными рисками. Экономически оправданная безопасность. М.: Компания АйТи; ДМК Пресс, 2005.

Архитектура ядра системы мониторинга и защиты от вторжений

С. С. Корт, Е. А. Рудина

Интеграция возможностей различных средств обнаружения атак, межсетевых экранов, средств аудита безопасности, прочих средств, контролирующих состояние защищенности системы, которая позволяет добиваться наиболее полного перекрытия внешнего негативного воздействия на систему, может выполняться следующими путями:

- автоматически, посредством взаимодействия компонентов в рамках единой системы;
- путем выполнения экспертного анализа результатов работы одних средств (совместно или по отдельности) и настройкой в соответствии с этими результатами других средств.

Очевидно, что первый подход является предпочтительным, поскольку позволяет организовывать управление функциями защиты системы в режиме реального времени, экономит время эксперта, облегчает администрирование системы. Недостатком этого подхода является необходимость создания такой многокомпонентной системы защиты для решения задачи обеспечения безопасности практически любой системы. Создание неуниверсальной системы, предназначенной для решения задачи обеспечения защиты в каждом отдельно взятом случае, как правило, не оправдано затратно. В результате, предлагаемые решения, как правило, не реализуют все множество функций защиты для каждого случая, однако являются более или менее универсальными.

Примером слияния функций двух различных средств защиты — систем обнаружения вторжений (СОВ) и межсетевых экранов — являются т.н. системы предотвращения вторжений (СПВ). В настоящее время на рынке присутствует множество программно-аппаратных продуктов и подключаемых к ним компонентов, реализующих различные функции СОВ и СПВ.

Современные системы обнаружения и предотвращения вторжений демонстрируют тенденцию к слиянию своих функций и традиционных функций других средств обеспечения безопасности. Сетевые СПВ объединяют функции сетевой СОВ и межсетевого экрана (МЭ). СОВ, коррелирующая

результаты своей работы с автоматически собираемыми ею значениями параметров защищаемой системы, выполняет функции пассивного аудита безопасности системы. Такие системы с расширенной функциональностью перестают быть СОВ и СПВ в традиционном понимании. Поэтому такую, как правило, многокомпонентную систему будем называть более общо — системой мониторинга и защиты от вторжений (СМЗВ). *Система мониторинга и защиты от вторжений* вычислительной системы реализует множество функций, направленных на контроль и предотвращение (различными способами и методами) нарушений безопасности.

Рассмотрим возможность создания архитектуры, позволяющей увеличивать расширяемость и гибкость функциональных возможностей системы мониторинга и защиты от вторжений путем ее сборки из отдельных модулей. При этом способы, методы и алгоритмы сбора данных, их обработки и анализа, определяющие, в конечном счете, функциональные характеристики готовой системы, выносятся из области рассмотрения. Требуемое решение должно представлять собой хорошо спроектированный, повторно используемый каркас СМЗВ, позволяющий достаточно быстро и эффективно внедрять как уже существующие, так и новые компоненты, реализующие возможности системы по обеспечению требуемой степени защиты (компонент защиты — некоторый модуль, имеющий хорошо определенную область ответственности и четко обозначенные множества входных и выходных данных, и в совокупности с другими такими компонентами реализующий функциональные возможности СМЗВ).

При этом можно выделить следующие преимущества решений на базе такой архитектуры:

1. Сокращение времени на разработку — возможность повторного использования универсальных базовых механизмов (ядра архитектуры) компонентами защиты при разработке этих компонентов.
2. Гибкость — возможность синтезировать системы из различных компонентов защиты с минимальными затратами на обеспечение совместимости этих компонентов.
3. Открытость — возможность внедрять дополнительные компоненты защиты в уже существующую систему с целью улучшения ее характеристик.

Поскольку описываемая СМЗВ является функциональным обобщением СОВ, для создания ее структуры рассмотрим в первую очередь классическую модель СОВ, предложенную в работе Д. Деннинг [1]. Обобщение структуры СОВ на основании данной модели было сделано в документе CIDF (Common Intrusion Detection Framework) [2].

Классическая модель СОВ не вполне адекватно описывает состав и архитектуру современных СОВ и СПВ. Рассмотрим причины несоответствия современных систем структуре, описанной в документе CIDF.

Современную СОВ/СПВ может составлять множество компонентов различных видов, не все из которых возможно описать как модули структуры, описанной в документе CIDF. Также некоторые виды компонентов по своему назначению могут быть отнесены к нескольким модулям структуры CIDF. Даже если рассматривать как модули структуры CIDF не как отдельные компоненты каждой конкретной системы, а как группы компонентов, не всегда возможно установить соответствие между реальной системой и этой структурой. Это связано как с расширением функциональности современных СОВ, так и с методами улучшения качества обнаружения.

Документ CIDF не предъявляет требований по обеспечению совместимости модулей, которые позволяют в современных системах использовать механизм встраиваемых компонентов, обеспечивающих гибкость выполняемых системой функций.

Структура, определенная в CIDF, не учитывает возможность выполнения анализа в несколько этапов на разных семантических уровнях, характерного для современных алгоритмов обнаружения. Системы, реализующие такие алгоритмы анализа, обычно включают в свой состав несколько компонентов анализа, причем некоторые из этих компонентов являются источниками входных данных для других. Это приводит к тому, что обмен данными в рамках системы не соответствует обмену данными, описанному в рамках CIDF.

В CIDF отсутствуют понятия управления, контроля и обеспечения собственной безопасности компонентов, составляющих систему.

Для современных СОВ и СПВ чрезвычайно актуально требование обеспечения системой самоконтроля, управления собственными компонентами и обеспечения безопасности своего функционирования.

Таким образом, классическая структура СОВ, приведенная в документе CIDF, оказывается недостаточной для описания многих СОВ и СПВ. Основная причина — неполный учет как предметной области, так и особенностей функционирования модулей системы.

Невозможность полного и непротиворечивого соотнесения структуры произвольной СМЗВ со структурой, определенной в документе CIDF, свидетельствует о необходимости пересмотра этой структуры.

Для решения задачи построения структуры произвольной многокомпонентной СМЗВ, с учетом проблем, описанных выше, в докладе предложен следующий подход. Выделение модулей структуры должно основываться не на их конкретном функциональном назначении (хранение данных, выполнение первичного или вторичного анализа и пр.), поскольку такой

подход не всегда применим к описанию, например, гибридных или качественно новых типов модулей системы, а на том, что эта часть имеет некоторую четко идентифицируемую (но не обозначаемую конкретно) область ответственности на некотором семантическом уровне интерпретации данных. Таким образом, получено самое общее представление структуры. Постепенное усложнение этой структуры на основании требования универсальности и особенностей существующих систем, позволяет перейти к представлению архитектуры СМЗВ.

Структура СМЗВ, отвечающая описанным условиям, приведена на рис. 1. Эта структура предоставляет возможность создания архитектуры, в рамках которой различные подключаемые через определенный интерфейс компоненты защиты смогут составлять индивидуальные решения задачи обеспечения безопасности конкретной системы.

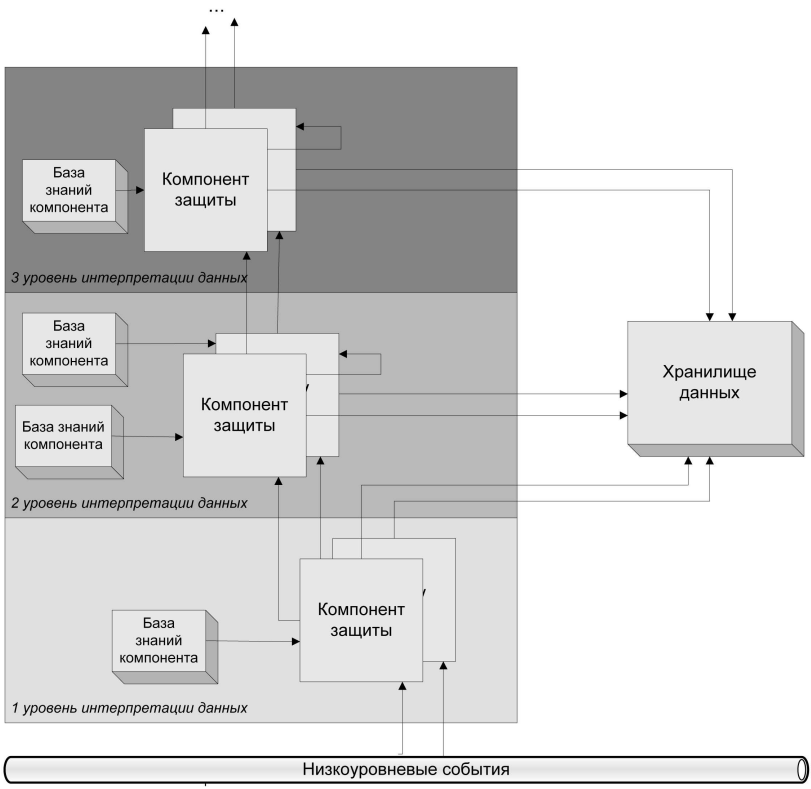


Рис. 1. Обобщенная структура СМЗВ

Приведенная структура СМЗВ позволяет решить следующие проблемы, сформулированные при описании структуры CIDF:

1. *Невозможность однозначного соотнесения некоторых типов компонентов защиты и модулей структуры CIDF*. Исключение из приведенной структуры типизации модулей позволяет применять эту структуру для описания сложных многокомпонентных СОВ с расширенными функциональными возможностями. Из всех модулей тип обозначен только для модуля хранения данных, являющегося центральным звеном любой такой системы.

2. *В CIDF отсутствуют требования по обеспечению совместимости модулей*. Увеличение гибкости структуры за счет исключения типизации позволяет, удовлетворив некоторые условия, обеспечить также ее расширяемость. А именно: если модуль хранения имеет достаточно универсальную схему хранения данных, относящихся к контролю защиты системы, и предоставляет интерфейс доступа к этим данным, то это позволяет обеспечить расширяемость функциональности системы путем создания дополнительных модулей, совместимых с уже существующими компонентами.

3. *Несоответствие потоков обработки информации современными системами структуре CIDF*. В приведенной структуре в сочетании со снятым ограничением на количество модулей, структуризация потоков данных в системе согласно семантическому уровню интерпретации этих данных позволяет описать различные методики обеспечения защиты.

4. *Отсутствие в CIDF требований к самоконтролю и обеспечению системой собственной безопасности*. Данная структура, как и структура CIDF, явным образом не специфицирует модули, непосредственно ответственные за обеспечение защиты самой системы. Но, как было отмечено, эти компоненты защиты могут рассматриваться как внешние по отношению к структуре, или, при необходимости, — как ее внутренние модули.

Основываясь на предложенной структуре, уточним требования к базовым механизмам СМЗВ и компонентам защиты. Для этого необходимо установить на основании анализа деталей архитектуры существующих СОВ и СПВ требования к архитектуре системы, выполнение которых необходимо для обеспечения универсальности этой системы.

Требования к хранилищу данных СМЗВ

Одной из наиболее важных проблем, которые необходимо решить при построении любой СОВ, является проблема представления и хранения

данных. Универсальность архитектуры подразумевает также независимость представления данных от интерпретации этих данных конкретным компонентом защиты, включаемым в систему (от алгоритма анализа этих данных, способа их вывода и пр.), и от формата выходных данных этого компонента.

Следовательно, модуль хранения и доступа к данным должен отвечать следующим требованиям:

- предоставлять достаточно ресурсов для хранения данных (результатов работы) любых компонентов защиты;
- хранить информацию о времени генерации данных и компоненте защите, сгенерировавшем данные;
- хранить информацию о корреляционных связях различных данных, возможно, полученных от различных компонентов защиты;
- хранить все данные в универсальном формате, который бы позволил любому компоненту защиты использовать данные любого другого компонента, если ему это необходимо;
- предоставлять компонентам защиты доступ к данным с использованием некоторого универсального интерфейса, скрывающего детали организации модуля хранения, и упрощающего получение данных из хранилища;
- разрешать сохранение результатов работы компонентов защиты также с использованием этого интерфейса, упрощающего вывод данных и гарантирующего корректное использование универсального формата.

Необходимость универсального формата данных обусловлена требованием обмена данными компонентами. Даже если различные компоненты описывают одни и те же данные различным образом, необходим некоторый универсальный формат, в который можно было бы экспортировать данные обоих компонентов.

Требование совместимости формата хранения данных и стандарта IDMEF

Существующим стандартом, призванным обеспечить интероперабельность коммерческих, свободно распространяемых и исследовательских СОВ и СПВ, является стандарт IDMEF. Модель данных IDMEF является объектно-ориентированной моделью, описывающей основные сущности в области обнаружения вторжений в виде классов, находящиеся в определенных отношениях наследования и включения. IDWG также предлагает

протокол обмена сообщениями в формате IDMEF — IDXP. Необходимо отметить, что сообщения IDMEF нецелесообразно использовать для хранения данных в силу соображений эффективности (каждое сообщение представляет собой документ XML).

При проектировании модуля хранения основной задачей является выделение необходимого и достаточного множества параметров, описывающих результаты работы практически любого компонента защиты. Множество параметров, значения которых должны сохраняться, включает в себя множество атрибутов классов, описываемых стандартом IDMEF. Кроме того, это множество может быть дополнено путем исследования уже существующих средств защиты (различных СОВ, СПВ, средств аудита безопасности, межсетевых экранов) с тем, чтобы включить в него параметры, не описываемые IDMEF, но необходимые для обеспечения универсальности.

При этом нужно отделить параметры, описывающие события, относящиеся к деятельности защищаемых объектов и источников угроз, от параметров собственно алгоритма оценки этих событий. К примеру, атрибуты одного и того же набора событий (на хосте или в сети) могут оцениваться на наличие сигнатур атак, с точки зрения выполнения предикатов из указанного набора, на предмет удовлетворения некоторым статистическим зависимостям и т.п. При этом хранилище базы данных сигнатур атак, набора предикатов или статистических профилей физически (по расположению) и логически (по интерфейсу доступа) должно быть разделено с хранилищем данных универсальной системы. Организация базы знаний для поддержки работы того или иного алгоритма интересует только компонент защиты, реализующий этот алгоритм, и совершенно неважна с точки зрения разработки универсальной архитектуры системы.

Требование организации доступа к хранилищу данных с использованием универсального интерфейса

Наиболее подходящее с точки зрения эффективности решение для организации хранилища СОВ или СПВ — использование базы данных. Однако затруднительно остановить выбор на конкретной СУБД, в силу нескольких причин. Во-первых, для каждого решения на базе универсальной архитектуры выдвигаются индивидуальные требования по скорости доступа к данным, расширяемости, защищенности; существуют и другие требования (поддержки СУБД на конкретной платформе, финансовые требования и т.п.). Во-вторых, организация доступа к данным со стороны компонентов защиты требует выполнения запросов на языке SQL, стандар-

тизованного, но имеющего практически для каждой СУБД свой диалект. В-третьих, непосредственное обращение компонентов защиты к данным, хранимым СУБД, может привести к вольностям и отклонениям в формате сохраняемых сообщений, что, в свою очередь, приведет к потенциальной несовместимости компонентов.

Решением является создание универсального программного интерфейса доступа к данным хранилища системы на языке программирования широкого применения. Интерфейс реализуется некоторой библиотекой, экспортирующей все необходимые функции и данные. Такой подход позволяет скрыть реализацию хранилища и упростить доступ к данным отдельных компонентов защиты, а также создавать СОВ и СПВ на базе различных СУБД, в зависимости от предъявляемых к ним требований.

Требование организации работы компонентов защиты в различных режимах и обеспечение их взаимодействия

Другим важным требованием является требование реализации механизма включения и организации взаимодействия отдельно взятых компонентов защиты в системе.

Рассмотрим возможные режимы запуска и работы компонентов защиты:

1. Компоненты, запускаемые на выполнение единственный раз и, возможно, продолжающие оставаться в фоновом режиме работы. Типичным примером такого компонента являются сетевые и хостовые датчики событий, межсетевые экраны, менеджеры блокировки хоста или процесса.
2. Компоненты, запускаемые на выполнение раз в указанный промежуток времени. Периодически может быть необходимо запускать средства сбора текущих параметров и аудита безопасности, средства пост-обработки, генераторы вторичных атрибутов и другие модули пост-обработки событий.
3. Компоненты, запускаемые при фиксации некоторого события другим компонентом этого же или, чаще, другого типа. Это могут быть модули анализа в режиме реального времени, модули уведомления. Данный режим позволяет организовать комплексную обработку события в режиме реального времени. Недостатком данного режима по сравнению с предыдущими является необходимость знания компонентов защиты друг о друге и их синхронизации.

Если практически такую синхронизацию организовать сложно, можно использовать режим периодического запуска, сканируя базу данных событий на предмет появления новых сообщений определенного типа. Период запуска должен быть достаточно малым, чтобы обработка осуществлялась в режиме, сравнимом с режимом реального времени (но не настолько малым, чтобы большинство запусков гарантированно были «холостыми»). Режим периодического запуска с точки зрения возможности имитации остальных режимов является наиболее универсальным.

Приведенное описание режимов запуска компонентов не учитывает возможность сбоя этих компонентов и может быть скорректировано с учетом необходимости восстановления после сбоя.

Система должна предоставлять интерфейс для подключения компонентов защиты различных типов и реализовывать возможность запуска каждого компонента в указанном им режиме и с указанными параметрами режима: единственный раз, раз в указанный период времени или при срабатывании другого компонента указанного типа. Тип компонента определяется областью его ответственности, методом, алгоритмом и соответствующей реализацией. Например, различные версии одного и того же программного продукта можно рассматривать как компоненты одного типа. Компоненты, декларирующие свою принадлежность к одному типу, должны продуцировать однотипные результаты (результаты, которые могут быть интерпретированы одинаковым образом). Тип компонента декларируется им при регистрации этого компонента. Также компонент при регистрации в системе должен указать режим своего запуска и, в случае запуска по событию, тип компонентов, от которых он зависит (при фиксации событий от которых он должен быть запущен). Информация о компонентах и их взаимосвязи хранится в области самоописывающих данных хранилища данных.

Для уже существующих систем и средств обеспечения безопасности можно создать компоненты импорта результатов их работы в хранилище данных СОВ. Также многие системы, в особенности свободно распространяемые, позволяют непосредственно создавать встраиваемые в них модули вывода. Такие компоненты импорта могут запускаться раз в тайм-аут, включая, таким образом, уже существующую систему в состав универсальной СМЗВ.

На основании общей структуры и приведенного выше анализа предъявляемых к СМЗВ требований, можно предложить ядро СМЗВ, составленное следующими базовыми механизмами (рис. 2):

- хранилище данных;
- реализация интерфейса доступа к хранилищу данных;

- процессор запуска и организации взаимодействия компонентов защиты, предоставляющий интерфейс их подключения.

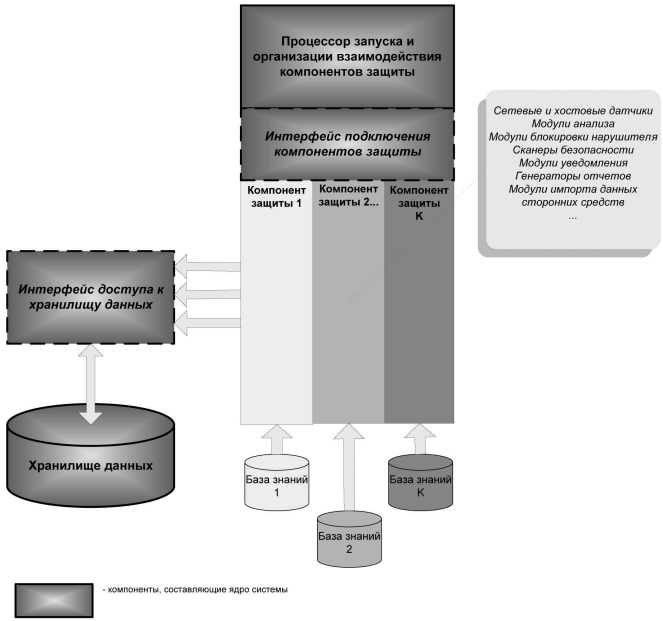


Рис. 2. Архитектура системы мониторинга и защиты от вторжений

Основываясь на экспортируемых системой интерфейсах этих базовых механизмах, подключаемые компоненты защиты формируют собственно функциональные возможности системы. Различные типы компонентов могут взаимодействовать между собой, инициируя запуск обработки данных по результатам работы другого компонента. Каждый из компонентов может использовать собственную, произвольным образом организованную базу знаний, поддерживающую работу его алгоритма. Таким образом, интерфейс подключения компонентов защиты обеспечивает возможность создания системы контроля безопасности с расширяемой функциональностью. Организация доступа к хранилищу данных посредством программного интерфейса позволяет увеличить гибкость и простоту реализации подключаемых компонентов защиты.

Описанная архитектура позволяет объединить функциональные возможности различных средств обеспечения безопасности в единую систему, организовать взаимодействие (в том числе, в режиме реального времени)

компонентов, имеющих различные области ответственности. Архитектура позволяет создавать индивидуальные решения для защиты компьютерных систем и сетей в рамках одной системы на базе универсального хранилища данных и процессора синхронизации компонентов, составляющих ядро системы. Модуль хранения и механизмы синхронизации реализуются при создании практически каждой более или менее крупной системы, направленной на обнаружение вторжений и обеспечение безопасности систем и сетей. Наблюдаемая тенденция к слиянию традиционных функций СОВ с функциями других средств обеспечения безопасности (межсетевых экранов, средств аудита безопасности и пр.) с целью универсализации назначения этих систем приводит к необходимости создания повторно используемого решения, коим и является предложенная архитектура системы контроля безопасности.

Литература

[1] *Denning D.* An intrusion-detection model // IEEE Transactions on Software Engineering. February 1987. V. Se-13, №. 2. P. 222–232.

[2] The Common Intrusion Detection Framework Architecture. Common Intrusion Detection Framework (CIDF) working group paper.

Задача оптимальной расстановки систем мониторинга потоков

О. Д. Соколова, А. Н. Юргенсон

1. Введение

Одна из задач мониторига безопасности информационной сети связана с оперативным отслеживанием состояния сети и ее компонентов с целью обнаружения аномального поведения, которое может быть следствием атак на сеть. В работе рассматривается задача оптимального расположения систем-«наблюдателей» на заданной информационной сети с виртуальными каналами.

2. Задача мониторинга потоков в сети

Рассмотрим сеть с заданной топологией, передача данных в которой осуществляется по виртуальным каналам. Возможны несанкцианированные вторжения, которые могут отрицательно влиять на работу сети, например, загрузка канала спамом, установка аппаратуры для передачи идущего трафика по ложному маршруту и ряд других. Необходимо расставить на первичной сети устройства (назовем их «наблюдатели») для обнаружения показателей аномальности в работе сети.

Устройства, работающие в узлах сети, отличаются и по набору функций, и по стоимости от аналогичных устройств для поддержки ее каналов. По этой причине здесь рассматривается задача расстановки устройств только на каналах сети.

Считаем, что задано ограничение на общую стоимость всех устройств — C, и требуется расставить «наблюдателей» так, чтобы, оставаясь в рамках ограничения на суммарную стоимость C, покрыть наибольшее число соединений.

3. Математическая постановка задачи поиска мест расположения «наблюдателей»

Наиболее часто используемыми математическими моделями при описании структур сетей являются графы и гиперсети [1].

Гиперсетью S называется шестёрка $S = (X, V, R; P, F, W)$, состоящая из следующих объектов:

$X = (x_1, x_2, \ldots, x_n)$ — множество вершин;

$V = (v_1, v_2, \ldots, v_m)$ — множество ветвей;

$R = (r_1, r_2, \ldots, r_k)$ — множество ребер.

$P : V \to 2^X$— отображение, сопоставляющее каждому элементу $v \in V$ множество $P(v) \subseteq X$ его вершин. Таким образом отображение P определяет первичную сеть $PS = (X, V)$;

$F : R \to 2^V$ — отображение, сопоставляющее каждому элементу $r \in R$ множество $F(r) \subseteq V$ его ветвей.

$W : R \to 2^X$ — отображение, сопоставляющее каждому элементу $r \in \in R$ множество $W(r)$ его вершин. Отображение определяет вторичную сеть $WS = (X, R)$.

Считаем, что первичная сеть задана в виде графа $G = (X, V)$, где X — множество вершин, $|X| = n$, V — множество каналов связи, $|V| = m$.

Каждый канал имеет две характеристики: длину и пропускную способность.

Вторичная сеть задана множеством тяготеющих пар X' , и на этом множестве возможны различные соединения между вершинами. Так как неизвестно, между какими вершинами из множества X' будет осуществляться связь, то будем считать, что на множестве X' задан полный граф (соединения берутся с избытком). В задаче требуется осуществить синтез гиперсети с учетом пропускных способностей каналов. Для ее решения для каждой пары вершин из X' ищется кратчайший маршрут для прохождения соединения по ветвям первичной сети с учетом ограничений на пропускные способности каналов. Используется модификация алгоритма Флойда.

После решения этой задачи имеем вложение вторичной сети в первичную. Таким образом задается гиперсеть. Для нее на ветвях первичной сети нужно найти места размещения «наблюдателей», чтобы все потоки (т. е. ребра вторичной сети) были под наблюдением.

В терминах теории гиперсетей — это задача поиска покрытия ветвями первичной сети всех ребер вторичной сети. Так как задача покрытия NP-полная, то разработан приближенный эвристический алгоритм. Чтобы найти минимальное покрытие, строим двудольный граф: вершины первой доли — ветви первичной сети, вершины второй доли — ребра вторичной сети. Если поток информации (вершина второй доли графа) проходит по

нескольким каналам (вершины первой доли), то существуют ребра в двудольном графе между этими вершинами. Далее ищем минимальное покрытие вершинами первой доли всех вершин второй доли.

С учетом того факта, что задано ограничение C, алгоритм работает до тех пор, пока общая стоимость размещенных «наблюдателей» не достигнет порогового значения, и, следовательно, не всегда обеспечивается полное покрытие всех потоков. Однако, принимая во внимание, что вторичная сеть была построена с избытком, а в реальной ситуации передача данных идет не по всем ребрам, то после работы алгоритма необходимо протестировать, как найденное покрывающее множество решает задачу покрытия конкретной вторичной сети. Для этого случайным образом удаляем из полного списка ребер вторичной сети несколько ребер и проверяем, какая доля из оставшихся оказалась покрыта «наблюдателями».

На рис. 1 показаны графики зависимости доли покрытых ребер вторичной сети (по оси OY) от количества «наблюдателей» (по оси OX). Структура первичной сети ($|X| = 50$, $|X'| = 10$) задана в виде звезды (1a), в виде простой цепи (1b), в виде цикла (1c) и в виде решетки 5×10 (1d).

Были протестированы различные графы: первичная сеть — случайный граф с количеством вершин до 500, количеством ветвей до 600. Время работы программы при этом не превышало нескольких секунд.

В [2] было показано, что в случае, когда существует точное решение, состоящее из двух ветвей, приближенный алгоритм, прекративший работу после достижения ограничения 2, находит решение, покрывающее не менее 75% ребер.

Для случая, когда оптимальное покрытие состоит из трех ветвей, приближенный алгоритм, остановленный после достижения ограничения 3, находит решение, покрывающее не менее 70% ребер.

Можно также решать и обратную задачу — входным параметром брать не суммарную стоимость «наблюдателей», а долю покрытых потоков. В этом случае по результатам работы программы можно определять количество «наблюдателей», которое необходимо для обеспечения мониторинга заданной доли потоков в сети.

Литература

[1] *Попков В. К.* Математические модели связности. Новосибирск:о ИВМиМГ, 2006.

[2] *Соколова О. Д.*, *Юргенсон А. Н.* Об одной задаче мониторинга информационных потоков // Труды ИВМиМГ СО РАН. Серия Информатика. Новосибирск, 2008. Вып. 8. С. 132–137. (Материалы Четвертой азиатской международной школы-семинара.)

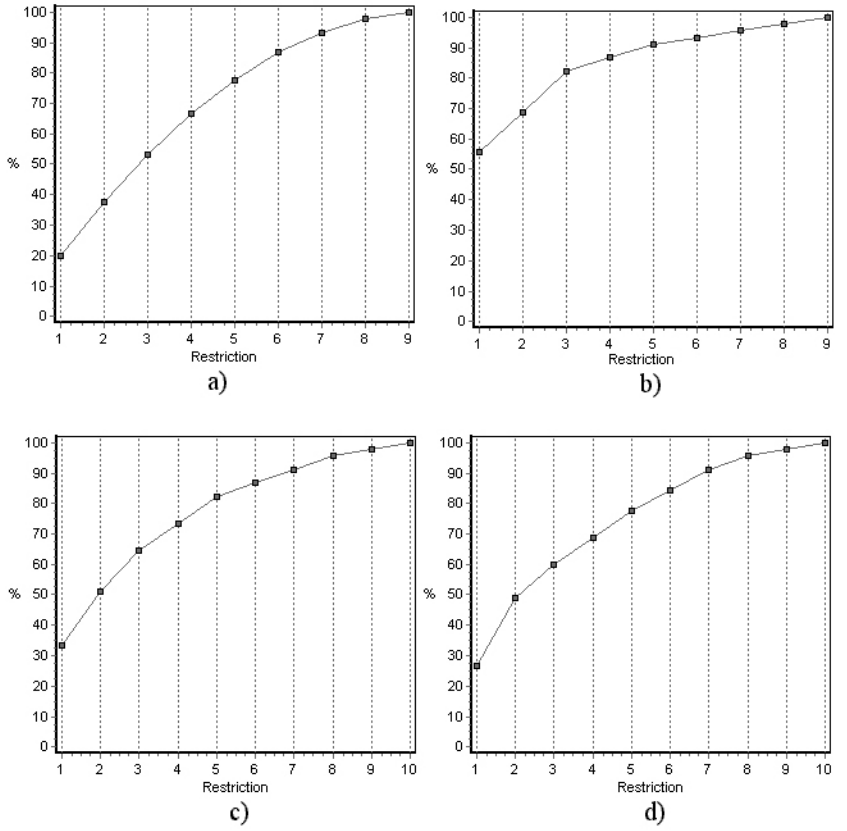


Рис. 1. Зависимость доли покрытых ребер вторичной сети от количества «наблюдателей»

[3] *Зима В.*, *Молдовян А.*, *Молдовян Н.* Безопасность глобальных сетевых технологий. СПб.: БХВ-Петербург, 2000.

[4] *Кравчук С. В.*, *Платонов В. В.* Методы обнаружения атак и системы обнаружения несанкционированных вторжений. `http://www.ssl.stu.neva.ru/ssl/publications/magazine/1999/4/6/kravchuk.pdf`.

Проектирование и анализ протокола удаленного доверия

В. А. Десницкий, И. В. Котенко

1. Введение

Работа посвящена исследованию модели защиты программ на основе механизма «удаленного доверия» в [1] и, в частности, разработке и анализу коммуникационного протокола (entrusting-протокола), предназначенного для обеспечения безопасной передачи сообщений в рамках данной модели защиты.

В работе рассматриваются основные виды атак на entrusting-протокол и соответствующие им модели нарушителя. Формируются основные требования к безопасности протокола. Предлагается общая методика построения entrusting-протокола, а также варианты реализации его программного прототипа.

2. Сущность механизма «удаленного доверия»

Основными элементами рассматриваемой модели защиты являются клиентская программа, которая выполняется на ненадежном хосте и подлежит защите, и удаленный сервер, функционирующий на надежном хосте. Программа подвержена атакам со стороны пользователей, которые хотели бы нарушить ее корректное выполнение.

Основная цель данного механизма защиты — гарантировать неизменность и корректность работы программы, работающей в потенциально враждебном окружении.

Одним из наиболее важных принципов рассматриваемого механизма защиты является внедрение в защищаемую программу специального переносимого (мобильного) модуля, представляющего собой программный компонент, в который входят монитор и генератор подписей.

Работа выполнена при финансовой поддержке РФФИ (проект № 07-01-00547), программы фундаментальных исследований ОНИТ РАН, Фонда содействия отечественной науке и при частичной финансовой поддержке, осуществляемой в рамках проекта Евросоюза RE-TRUST (контракт № 021186-2) и других проектов.

В задачу монитора входит постоянное осуществление проверок программы во время ее выполнения. Такие проверки включают верификацию бинарного кода, текущего состояния программы, выполняющихся процессов, версий используемых библиотек и другие подобные им действия. В качестве проверок могут также выступать специальные контрольные вычисления, чувствительные к модификациям программы. В случае злонамеренной модификации результат такого вычисления будет отличаться от ожидаемого, что даст возможность серверу ее обнаружить. Результаты верификации проходят стадию шифрования, после чего на их основе генерируются цифровые подписи, которые отправляются серверу.

Важнейшим требованием, предъявляемым к подписям, является сложность их анализа злоумышленником. Необходимо также отметить требование уникальности подписей, как для каждого клиента, так и для каждого запуска программы в рамках одного клиента.

Отличительной особенностью модуля является то, что он не поставляется совместно с программой, а загружается с надежного хоста во время первого запуска программы и динамически встраивается в нее, и далее регулярно обновляется в определенные моменты времени. Замена модуля производится для повышения его устойчивости к взломам. Основное требование к процессу замещения состоит в том, чтобы расчетное время, которое злоумышленник будет затрачивать на взлом мобильного модуля, не уменьшалось бы ниже определенного значения, даже при условии, что противник смог взломать все предыдущие версии модуля.

При обнаружении вмешательств доверенный сервер прекращает рассматривать клиента в качестве надежного и, соответственно, прекращается предоставление ему всех доступных сервисов, программных обновлений и других видов сопровождения.

Разработка и анализ специализированного протокола обмена сообщениями («entrusting-протокола»), как составной части механизма защиты, является задачей, требующей отдельного рассмотрения. Entrusting-протокол предназначается для защищенного обмена сообщениями между клиентской программой и доверенным сервером, и, в частности, для доставки кода мобильного модуля и данных, содержащих результаты выполненных проверок.

3. Типы атак и модели нарушителя

Рассмотрим два следующих далее основных типа атак на entrusting-протокол.

- Прослушивание коммуникационных каналов, по которым осуществляется передача данных. Такая атака сама по себе не представляет

серьезной угрозы для корректной работы entrusting-протокола, однако является необходимым условием для осуществления других атак на механизм защиты и клиентскую программу.

- Злонамеренное изменение передаваемых и обрабатываемых в рамках соответствующего клиентского компонента данных.

В случае, когда модификации подвергаются отправляемые клиенту данные (код мобильного модуля), целью такой атаки (подмены модуля) является намерение повлиять на программу извне с целью изменения ее поведения. Основной же целью такой атаки является намеренное искажение данных о текущем состоянии клиентской программы, которую получает и затем анализирует доверенный сервер.

Рассмотрим две модели нарушителя, характеризующие возможные действия злоумышленника по компрометации entrusting-протокола [2]. Первая из них позволяет описывать и анализировать возможные атаки типа «man-in-the-end» («человек на конце»), тогда как вторая модель ориентирована на атаки «man-in-the-middle» («человек посередине»). На практике злоумышленник комбинирует эти атаки.

В первом случае предполагается, что вмешательствам подвергается специализированный программный компонент, который реализует клиентскую сторону entrusting-протокола и может также находиться в пределах мобильного модуля. Такое злонамеренное изменение осуществляется атакующим для модификации правил работы протокола, и, в частности, правил формирования исходящего клиентского трафика. В случае, если entrusting-протокол является достаточно стойким, то изменения подобного вида могут быть обнаружены доверенным сервером опосредованно, как обнаружение какой-либо несогласованности в протоколе или при обнаружении отклонения значений полученных данных от некоего ожидаемого набора значений. Частным случаем такой атаки является прослушивание ключей, которые в явном виде не передаются по сети, а формируются на стороне клиента.

Атаки man-in-the-middle представляют прослушивание или захват трафика на пути его следования между клиентом и сервером. В качестве субъекта атаки может выступать как непосредственно конечный пользователь клиентской программы, пытающийся вмешаться в сеанс протокола, так и некоторая третья сторона. Наиболее интересным является первый случай, где целью нарушителя является вмешательство в работу программы со стороны запускающего ее злонамеренного клиента.

Выделение двух рассмотренных моделей обуславливается необходимостью точного описания возможностей и целей потенциального нарушителя в контексте entrusting-протокола. Заметим, что разные реализации атак

могут оказываться для нарушителя предпочтительными в различных ситуациях. К таким условиям можно отнести: сложность обнаружения атаки доверенным сервером; сложность выполнения атак нарушителем на клиентской стороне или посередине; возможности автоматизации атаки.

4. Общая методика построения протокола

Основными требованиями к entrusting-протоколу, реализация которых позволяет утверждать о его адекватности целям исследований, являются аутентификация сторон протокола, согласование криптографических ключей, конфиденциальность передаваемых данных, аутентификация передаваемых данных и своевременность доставки данных.

В общем случае задача построения протокола, реализующего принцип удаленного доверия (как и любого другого протокола), представляет собой процесс, который включает:

- формирование целей и требований к безопасности протокола;
- синтез протокола на основе комбинирования отдельных протоколов, алгоритмов и криптографических примитивов;
- верификацию данной комбинации.

В результате, в случае обнаружения каких-либо уязвимостей протокол должен быть скорректирован, после чего должна быть произведена повторная его верификация.

Требования к безопасности могут быть составлены в виде онтологии, представленной иерархической структурой (дерево требований). В рамках такой структуры каждый узел представляет собой определенное требование безопасности. Такой узел, в свою очередь, может иметь дочерние узлы, отвечающие за более детальные требования, на которые может быть разложено данное требование, и уточняющие его.

Каждому конечному требованию (листу дерева требований) ставится в соответствие набор протоколов, алгоритмов и криптографических примитивов, посредством которых оно может быть достигнуто. В общем случае каждый такой набор может состоять из достаточно большого числа средств, каждое из которых имеет свои свойства, отражающие определенные структурные и функциональные особенности. Таким образом, могут быть представлены различные стратегии по выбору конкретного средства для реализации заданного требования entrusting-протокола. В частности, на искомый протокол могут накладываться различные ограничения, например использование криптографических ключей, длина которых не пре-

вышает определенного значения, с целью уменьшить вычислительные издержки работы протокола.

Поскольку каждый из подходов к верификации имеет как свои достоинства, так и недостатки, для более полного представления о свойствах протокола в настоящей работе предполагается использовать несколько методов на основе применения систем AVISPA и Isabelle. Использование нескольких средств верификации, основанных на различных парадигмах, позволяет улучшить качество верификации протокола и, тем самым, повысить вероятность нахождения трудно обнаружимых уязвимостей в протоколе.

5. Реализация протокола

В работе анализируются три предполагаемые реализации программного прототипа entrusting-протокола. Протокол может строиться непосредственно на основе протокола TCP/IP (или UDP), который не имеет встроенных средств защиты. Данный прототип будет достаточно независимым с точки зрения реализации в конкретных сетях, и, как следствие, является адаптируемым (переносимым).

Два других прототипа основываются на использовании более сложных протоколов IPSec и TLS, способных обеспечить часть требований безопасности, а также на применении других средств, позволяющих реализовать дополнительные требования. К таким средствам можно отнести Time-Stamp Protocol (TSP), протокол, обеспечивающий реализацию меток времени, который основан на использовании X.509 сертификатов. Данный протокол предоставляет доказательство того, что некоторые данные были созданы не позднее определенного момента времени и не были модифицированы впоследствии. Возможным применением TSP является сертификация по времени подписей, отправляемых клиентом доверенному серверу.

6. Заключение

В работе исследована модель защиты программ на основе механизма «удаленного доверия», предложена методика разработки протокола, предназначенного для обеспечения безопасной передачи сообщений в рамках данной модели защиты, а также варианты реализации его программного прототипа.

В дальнейшем предполагается проведение углубленного анализа entrusting-протокола, в том числе его верификации на основе существующих

методов формального доказательства. Предполагается также исследование применимости методов автоматической генерации протоколов к рассматриваемой задаче, которые позволяют осуществлять построение протоколов таким образом, чтобы их корректность обеспечивалась на стадии их построения (correct-by-construction).

Литература

[1] *Ceccato M.*, *Ofek Y.*, *Tonella P.* Remote entrusting by run-time software authentication // SOFSEM 2008. Conference on Current Trends in Theory and Practice of Computer Science, Tatras, Slovakia, January, 2008.

[2] *Cederquist J.*, *Dashti M. T.* An intruder model for verifying liveness in security protocols // Proceedings of the fourth ACM workshop on Formal methods in security. Alexandria, Virginia, USA. 2006.

Автоматизация процесса мониторинга информационной безопасности компьютерных систем на основе политик безопасности

А. И. Тупицын

1. Введение

В настоящее время разработано достаточно много средств мониторинга информационной безопасности компьютерных систем. Большинство из них обладает широкими возможностями, позволяющими контролировать практически любой аспект функционирования компьютерной системы. Однако такие возможности порождают ряд вопросов связанных с настройкой параметров этих средств и контроля выполнения заданных свойств компьютерной системы на основании анализа данных мониторинга. С одной стороны, пользователи средств мониторинга стремятся установить наибольшее количество правил мониторинга для полного контроля компьютерной системы. С другой стороны, такие установки приводят к получению пользователями слишком большого количества данных мониторинга, что делает невозможным их исчерпывающий анализ. Нахождение разумного компромисса между стремлением собрать как можно больше информации и необходимостью её подробного анализа является достаточно трудной задачей. Даже в случае нахождения такого компромисса, количество данных, которые получаются в ходе мониторинга, существенно затрудняет их анализ.

Для разрешения указанного выше противоречия предлагается автоматизировать процесс мониторинга информационной безопасности компьютерных систем на основе политик безопасности. Политика безопасности организации (в настоящей работе — компьютерной системы) на верхнем уровне ее формализации представляет собой некоторое количество правил, процедур, практических приёмов или руководящих принципов в области безопасности, которыми руководствуется организация в своей деятельности [1].

2. Требования к средствам мониторинга информационной безопасности компьютерных систем, основанным на политиках безопасности

Ключевым назначением средств мониторинга информационной безопасности компьютерных систем, основанных на политиках, является повышение защищённости функционирования этих систем. Основными требованиями, предъявляемыми к средствам такого рода, выступают централизация и высокоуровневость [8].

Под централизацией понимается возможность задания всех необходимых параметров средства мониторинга в одном месте, вместо независимой настройки параметров каждой составной части этого средства. Под высокоуровневостью понимается возможность задания востребованных администраторами свойств функционирования компьютерной системы на уровне бизнес-процессов организации, а не на уровне деталей конкретных технологий, необходимых для обеспечения выполнения этих свойств.

В качестве дополнительного требования к средствам рассматриваемого класса можно указать требование принуждения к исполнению. Это требование означает наличие в средствах рассматриваемого класса механизмов, позволяющих принудить к исполнению принятого этими средствами решения по управлению информационной безопасностью компьютерной системы. Такие решения могут приниматься на основании как правил мониторинга, заданных пользователем, так и результатов анализа происходящих в компьютерной системе процессов.

Достаточно важным требованием к средствам рассматриваемого класса является доступность исходных кодов. Выполнение этого требования необходимо для обеспечения возможности применения средств мониторинга в тех компьютерных системах, в которых обязательно проведение исследования указанных выше средств на отсутствие недекларированных возможностей.

Сравнение основных средств мониторинга компьютерных систем, основанных на политиках, с точки зрения указанных выше характеристик приведено в таблице 1. Как видно из этой таблицы, в настоящее время не существует средства рассматриваемого класса, удовлетворяющего всем предъявляемым требованиям. В рамках настоящей работы рассматривается подход к построению такого средства путём автоматизации технологического процесса эксплуатации средства мониторинга.

Таблица 1. Сравнение характеристик средств мониторинга компьютерных систем, основанных на политиках

Требование / Средство	Централизация	Высокоуровневость	Принуждение к исполнению	Доступность исходных кодов
CORE FORCE [3]	Нет	Да	Да	Да
OSSIM [5]	Да	Да	Нет	Да
SunXACML [7]	Нет	Да	Нет	Да
POSITIF [6]	Да	Да	Да	Нет
IBM Tivoli [4]	Да	Да	Да	Нет
HP Software [2]	Да	Да	Да	Нет

3. Автоматизированный технологический процесс эксплуатации средства мониторинга компьютерных систем, основанный на политиках безопасности

В рамках решения задачи автоматизации процесса мониторинга информационной безопасности компьютерных систем, основанной на политиках безопасности, предлагается разработать технологический процесс эксплуатации средства мониторинга, обеспечивающий:

- формальную запись политик безопасности;
- автоматическое конфигурирование операционных систем и модулей средства мониторинга в соответствии с заданными политиками безопасности;
- контроль выполнения заданных политик безопасности.

Для решения указанной выше задачи предлагается выделить в процессе работы средства мониторинга следующие этапы.

1. Политики безопасности задаются в некотором документе и согласовываются как с подразделением, обеспечивающим безопасность, так и с эксплуатирующим подразделением.
2. Администратор безопасности с помощью графического интерфейса осуществляет формализацию и запись политик безопасности в средство мониторинга.
3. По команде администратора безопасности автоматически изменяется конфигурация операционных систем и модулей средства мониторинга в соответствии с заданными политиками безопасности.

4. В процессе эксплуатации средство мониторинга осуществляет:
   - контроль действий пользователей и инициированных ими процессов;
   - выявление несоответствия контролируемых действий заданным политикам безопасности;
   - контроль целостности настроек сенсоров и конфигурации операционных систем.

Графически автоматизированный технологический процесс эксплуатации средства мониторинга, основанный на политиках безопасности, представлен на рисунке 1.

Рис. 1. Автоматизированный технологический процесс эксплуатации средства мониторинга, основанный на политиках

Использование при эксплуатации средства мониторинга предлагаемого технологического процесса позволит обеспечить выполнение свойств:

- централизации — поскольку все конфигурационные параметры задаются в одном месте на основании политик безопасности;

- высокоуровневости — поскольку политики безопасности задаются на уровне бизнес-процессов организации;
- принуждения к исполнению — поскольку в процессе контроля действий пользователей и инициированных ими процессов неразрешённые политиками действия могут быть запрещены.

Для обеспечения выполнения требования доступности исходных кодов разработку средства мониторинга предлагается осуществлять с использованием программных средств с открытым исходным кодом и продуктов собственной разработки.

4. Выводы

В настоящей работе предложен автоматизированный технологический процесс эксплуатации средства мониторинга компьютерных систем, основанный на политиках безопасности.

В рамках работы были проанализированы существующие средства мониторинга, основанные на политиках, и показано, что ни одно из рассмотренных средств не удовлетворяет в совокупности требованиям централизации, высокоуровневости, принуждения к исполнению и доступности исходных кодов. По этой причине с целью обеспечения выполнения перечисленных требований разработан автоматизированный технологический процесс эксплуатации средства мониторинга.

На основании проведённой работы, можно сделать вывод о том, что предложенный подход к автоматизации технологического процесса эксплуатации средств мониторинга обеспечит выполнение предъявляемых к ним требований.

Литература

[1] ГОСТ Р ИСО/МЭК 15408–1–2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель.

[2] Решения для оптимизации бизнес-технологий (BTO). Available at `http://www.hp.ru/software`, November 2008.

[3] CORE FORCE — First Community-Oriented Security Solution for Personal Computers. Available at `http://force.coresecurity.com`, November 2008.

[4] IBM Tivoli Software — Russia. Available at `http://www-01.ibm.com/software/ru/tivoli/`, November 2008.

[5] OSSIM — Open Source Security Information Management. Available at `http://www.ossim.net`, November 2008.

[6] POSITIF Project. Available at `http://www.positif.org`, November 2008.

[7] Sun's XACML Implementation. Available at `http://sunxacml.sourceforge.net`, November 2008.

[8] *Verma D. C* Simplifying Network Administration using Policy based Management // IEEE Network. March/April 2002. V. 16, № 2. P. 20–26.

Исследование проактивных механизмов обнаружения вредоносного программного обеспечения на базе методов Data Mining

Д. В. Комашинский, И. В. Котенко

1. Введение

Несмотря на все усилия, прилагаемые различными научными коллективами и коммерческими компаниями, проблема защиты от вредоносного программного обеспечения (ПО) информационных ресурсов становится все более острой.

Актуальность обнаружения факта наличия на хосте функционирующих экземпляров вредоносных программ определяется, в первую очередь, смещением акцента с функциональности современного вредоносного ПО на увеличение скорости, скрытности, инвариантности его проникновения на атакуемый хост и длительности периода его функционирования. Важность решения этой задачи обусловлена также недостаточной эффективностью методов статического анализа потенциальных контейнеров вредоносного кода.

Оправданность задачи обнаружения вредоносного ПО в его активной фазе обусловлена тем обстоятельством, что именно в момент своего выполнения приложение оказывается наиболее «открытым» по отношению к наблюдателю, так как оно должно выполнять свои основные вредоносные функции. Кроме того, следует учитывать, что реализация поведенческого полиморфизма значительно сложнее создания структурного, что также должно учитываться в исследованиях в области детектирования вредоносного ПО.

В работе рассматривается подход к проактивному обнаружению вредоносного ПО, базирующийся на скрытном сборе информации о поведении запущенных приложений и ее обработке метода-

Работа выполнена при финансовой поддержке РФФИ (проект № 07-01-00547), программы фундаментальных исследований ОНИТ РАН, Фонда содействия отечественной науке и при частичной финансовой поддержке, осуществляемой в рамках проекта Евросоюза RE-TRUST (контракт № 021186-2) и других проектов.

ми интеллектуального анализа данных (*Data Mining*). Предлагаемый подход отличается от существующих направленностью на циклическую интерактивную скрытную обработку поведенческой информации, а также интегрированным использованием методов интеллектуального анализа данных для различных классов вредоносного ПО.

2. Релевантные работы

Вопрос о применимости средств Data Mining для решения задач обнаружения вредоносного ПО возник тогда, когда наступило осознание необходимости расширения мощности средств детектирования эвристическими методами. Не являясь панацеей, они вносят системный характер в процесс детектирования вредоносных программ по определенному экспертами набору признаков. В [1], например, рассматриваются возможности использования классификаторов Ripper, Naive Bayes и искуственного классификатора, построенного на мультипликативном объединении результатов работы нескольких классификаторов Naive Bayes. В качестве исходных данных для обучения используются данные, получаемые разбором структуры исполняемых файлов (строки из раздела ресурсов, данные таблиц импорта, бинарные фрагменты данных, элементы заголовков, содержащие данные о конфигурации приложения).

В [2] проведена оценка применимости классификаторов на основе деревьев решений и теоремы Байеса при использовании информации о наличии тех или иных коротких бинарных последовательностей в теле анализируемых файлов.

В [3] освещены вопросы создания системы детектирования опасных приложений на основе анализа их поведения. Исходными данными для обучения и верификации корректности работы бинарного классификатора, основанного на использовании метода опорных векторов, явились сгруппированные последовательности вызовов функций системных библиотек операционной системы.

3. Сущность предлагаемого подхода

Предлагаемый подход базируется на комплексном использовании методов Data Mining. Очевидно, что детектирование вредоносного программного обеспечения может основываться на двух основных подходах: на обнаружении заведомо опасных статических и поведенческих признаков (как правило, различных для разных классов вредоносного ПО) и на обнаружении аномальных признаков (отличающихся от типичных заведомо без-

опасных приложений). Именно особенности данных подходов указывают на необходимость проведения комплексных исследований в данной области. Необходимо иметь четкие оценки применимости тех или иных правил выделения признаков, отбора из их числа наиболее значимых и, в конечном итоге, методов Data Mining.

Следует отметить, что в целом задача детектирования вредоносного ПО имеет ряд критических требований, соблюдение которых является обязательным: (1) минимальные значения ошибок первого и второго рода принятия решения о вредоносности приложений, не находившихся в обучающем наборе; (2) скрытность сбора данных, необходимость которого обусловлена наличием разнообразных техник его обнаружения и противодействия ему; (3) удовлетворенность пользователя (своевременность и оперативность принятия решения). В рамках представляемого подхода, как будет показано далее, акцент смещен на обнаружение заведомо опасных поведенческих признаков. Он основан на использовании методов классификации, а именно — на отнесении объектов к тому или иному классу на базе формируемой математической модели.

Использование методов классификации предполагает проведение предварительного обучения выбранного классификатора с последующим использованием определенного набора настроенных весов. Признаки выделяются в процессе обучения классификатора на выборке, содержащей приложения, отнесенные к целевым классам. Пространство признаков является многомерным и определяется количеством выделенных признаков. Решающая математическая модель представляет собой функцию, определенную на пространстве признаков, оптимально разделяющую вектора, отнесенные к тому или иному классу на этапе обучения.

Используемые математические модели классификации основаны на использовании следующих групп методов [4]: статистических, в которых используются Naive Bayes и его специализации, уменьшающие влияние изначального предпожения о взаимной независимости аттрибутов; индуктивных, когда применяются деревья решений; классификаторов, базирующихся на основе разделимости множеств. Отметим, что на данной фазе исследований был использован классификатор на основе многослойного перцептрона.

Сбор исходных данных в подходе основан на мониторинге вызовов низкоуровневых функций операционной системы. Это позволяет получать хронологически корректную последовательность о фактах использования приложением критических системных ресурсов, без которых сложно создать полноценный вариант работоспособного вредоносного ПО.

Описание терминального инцидента (события) включает в себя: имя вызванной функции; значения переданных на вход функции операндов;

значения возвращенных функцией результатов. Формирование пространства признаков с учетом последних двух наборов и факт мониторинга низкоуровневых функций ОС выгодно отличает предлагаемый подход от описанного в [3]. Процесс выделения из набора хронологически упорядоченных инцидентов неоптимального множества признаков учитывает: количество вызовов каждой функции; количество обращений к ресурсам, обладающим одинаковым рангом значимости; факты запросов характерных ресурсов (попытки обращения к разделам системного реестра, файловой системе и подобных им); наличие и количество определенных цепочек вызовов.

Скрытность мониторинга основывается на намеренной модификации структур ядра операционной системы, недоступных для приложений, функционирующих в пользовательском режиме. Тем самым, функционирующее приложение лишено возможности по явным признакам вынести вердикт о наличии факта слежения за ним. Выполнение требований устойчивости и оперативности в общем случае зависит от качества программной реализации модулей перехвата и анализа. Кроме того, выполнение требования оперативности опосредованно определяется особенностями используемых классификаторов (например, возможностью их быстрого переобучения).

4. Комплекс моделирования и эксперименты

Программный комплекс сбора данных и оценки результатов построен на базе Windows XP (NT5.1). Сбор данных поведенческой информации основан на внедрении программных перехватчиков функций Native API. Для формирования исходных данных и данных для проверки сформированных классификаторами моделей используется набор вредоносных приложений из [5] и типовых безопасных приложений, входящих в состав операционной системы.

Для проведения обучения, кросс-проверок, контрольных проверок и визуализации результатов использовался специализированный программный комплекс Weka Classifier, распространяемый по условиям GNU General Public Licence. Запуск вредоносных программных приложений, необходимый для получения «трасс» их выполнения, производится в изолированной вычислительной среде под максимально привелегированной учетной записью.

Для проведения экспериментов была сформирована выборка вредоносных приложений, реализующих свой жизненный цикл с помощью функциональных рутин файлового перебора, сохранения резервных копий и автоматического запуска.

Результаты первой итерации работ, в том числе анализ промежуточных результатов, прояснили как некоторые вопросы оценки эффективности использования методов Data Mining в контексте темы данного исследования, так и направления дальнейшего расширения программного комплекса моделирования. К числу выводов, которые уже можно сделать относятся следующие: доступные исходные данные (файлы вредоносного ПО) имеют достаточно низкий уровень качества, не позволяющий без дополнительных усилий сформировать релевантную обучающую/тестовую выборку, которая привела бы к четкому и обоснованному результату. Собранные и проанализированные трассы выполнения вредоносного ПО подтверждают применимость более простых технологий детектирования, ориентированных на выбранный для начальных опытов класс вредоносных программ (контроль выделенных областей системного реестра, механизмов межпроцессного взаимодействия, работы с файловой системой). На начальном сформированном пространстве признаков и текущем (нерелевантном) тестовом наборе используемые классификаторы демонстрируют возможность обнаружения до 80% неизвестного вредоносного ПО при 15% показателе ложных срабатываний. Используемая группа статистических классификаторов в большей мере удовлетворяет требованиям оперативности в силу возможности инкрементальной дообучаемости.

Заключение

В работе предложен подход к проактивному обнаружению вредоносного ПО. Указанный подход позволяет осуществлять детектирование вредоносного программного обеспечения во время его выполнения за счет классификации по выделенным признакам его поведения. Вместе с тем, текущие результаты экспериментов показали необходимость более точного выделения наборов поведенческих признаков, которые характерны для каждого класса вредоносного ПО, необходимость расширения программного комплекса моделирования и его усложнения за счет учета дополнительных факторов.

Литература

[1] Schultz M. G., Eskin E., Zadok E., Stolfo S. J. Data Mining Methods for Detection of New Malicious Executables // Informatics and Computer Science, Volume 172, Issue 1-2, 2005, p. 241–261.

[2] Wang J.-H., Deng P. S., Fan Y.-S., Jaw L.-J., Liu Y.-C. Virus Detection using Data Mining Techniques // Proceedings. IEEE 37th Annual 2003 International Carnahan Conference, 14–16 Oct. 2003, p. 71–76.

[3] Zhang B.-Y., Yin J.-P., Hao J.-B., Zhang D.-X., Wang S.-L. Using Support Vector Machine to Detect Unknown Computer Viruses // International Journal of Computational Intelligence Research, Vol.2(1),2006, p. 100–104.

[4] Cios K. J., Pedrycz W., Swiniarski R. W., Kurgan L. A. Data Mining. A Knowledge Discovery Approach. Springer Science & Business Media, 2007.

[5] VX Heavens Site, `http://vx.netlux.org/`.

Формальное представление сетевого протокола

П. Д. Зегжда, Е. А. Рудина

1. Постановка задачи создания формального представления сетевого протокола

Создание средств обеспечения безопасности информации на сетевой среде, как правило, приводит к необходимости производить разбор сетевых протоколов с целью восстановления последовательности генерируемого ими обмена сообщениями. Для этого необходимо выполнить следующие функции:

- выделение и разбор отдельных сообщений, относящихся к реализации указанного протокола, из сетевого трафика;
- формирование текущего временно́го контекста в соответствии с разбираемыми пакетами и их метками времени;
- определение, является ли каждое из выделенных сообщений, корректным в текущем контексте обмена сообщениями;
- определение значений отдельных свойств выделенных сообщений.

Функции и процедуры, необходимые для восстановления последовательности обмена сообщениями и установления значений отдельных свойств различных сетевых протоколов (как показывает практика создания таких функций) весьма однотипны. В этой связи представляют интерес ответы на следующие вопросы: возможно ли создать обобщенный алгоритм восстановления последовательности обмена сообщениями?, каковы параметры этого алгоритма?

Если упомянутый обобщенный алгоритм существует, то описание произвольного сетевого протокола (для последующего его анализа) сводится к декларативному заданию характеризующих протокол параметров и соотношений. Объем кода, описывающего процедуру анализа протокола, и сложность его разработки существенно уменьшаются, поскольку функциональная часть анализа реализуется с помощью обобщенного алгоритма.

В рамках данной статьи поставлены и решены следующие задачи:

- описания формального декларативного представления произвольного сетевого протокола;
- создания обобщенного алгоритма для восстановления последовательности обмена сообщениями по некоторому сетевому протоколу, основанного на этом представлении;
- доказательства достаточной выразительности полученного математического аппарата для описания произвольного сетевого протокола;

2. Неформальное описание представления произвольного сетевого протокола

Декларативное описание произвольного сетевого протокола, в его гипотетически обобщенном представлении, подразумевает задание

- набора параметров обмена сообщениями;
- правил вычисления некоторого набора предикатов, определенных на параметрах обмена сообщениями, истинное или ложное значение которых определяет условные переходы алгоритма.

Значения одних параметров определяются только текущим сообщением в последовательности обмена. Значения других параметров определяются всей историей сообщений, частью этой истории, или они независимы от любого сообщения. Первый тип параметров будем называть свойствами сообщения, второй тип — свойствами контекста. Свойство сообщения (равно, как и свойство контекста) определяется как значение функции на множестве значений блока данных, описывающего сообщение, и, возможно, на декартовом произведении его с множествами значений других указанных свойств сообщения и/или контекста. Таким образом, функция свойства сообщения/контекста может определяться рекурсивно через другие функции свойств. Множество значений функции определяется для каждого свойства пакета/контекста индивидуально.

Пример свойств:

- длина пакета Len: $Packet \rightarrow [MinLength, MaxLength]$;
- версия протокола Ver: $Packet \rightarrow \{ValidVer1; ValidVer2; ValidVer3\}$;
- текущее состояние запроса (свойство контекста) Req: $Packet \rightarrow \{true, false\}$;

- корректность синтаксиса *Syntax*: *Packet X Ver_value X IsReq_value* →{*true*, *false*}.

Принятие решения об отнесении того или иного пакета (сообщения) к данному протоколу не должно зависеть от контекста. Данный предикат опирается только на значения свойств пакета. При принятии же этого решения возникает много предусловий дальнейшего разбора сообщения — контекст.

Рассмотрим связь различных свойств пакета. Можно выделить следующие свойства.

- Атомарные свойства. Эти свойства определяются только по данным пакета Packet и не зависят от других свойств. Примерами могут служить:
  - значение поля, находящегося по абсолютному смещению 0 от начала пакета (версия IP);
  - длина пакета;
  - корректный синтаксис с точки зрения четности числа кавычек, равного числа закрывающих и открывающих скобок и тому подобное;
- Свойства, выделяемые в зависимости от значения других свойств. При этом другие свойства могут быть как свойствами самого пакета (атомарными или зависимыми), так и свойствами контекста. Примерами могут служить:
  - поля, находящиеся по абсолютному смещению >0, в зависимости от корректности синтаксиса или длины пакета;
  - поля, смещение которых определяется другими полями;
  - поля кодов возврата, в зависимости от предшествующего пакета запроса;

Основной вопрос при определении сложности обобщенного представления протокола состоит в том, как могут коррелировать при анализе:

- различные свойства одного сообщения;
- свойства сообщения и свойства контекста.

3. Формализация описания произвольного сетевого протокола

Опишем математический аппарат представления протоколов, формализующий описанные выше абстракции. Пусть

- $P = \{prop\}_{1:n}$ — множество свойств сообщения;
- $A = \{aprop\}_{1:m}$, $m \leqslant n$, $A \subseteq P$ — множество атомарных свойств сообщения;
- $V_i = \{v_i\}_{1:k}$ $\forall i \in 1:n$ — множество значений свойства $prop_i$ (если $v_{is} = \varepsilon$, то значение свойства не определено);
- $C = \{ctx\}_{1:p}$ — множество свойств контекста;
- $U_j = \{u_j\}_{1:l}$ $\forall, j \in 1:p$ — множество значений свойств контекста (если $v_{js} = \varepsilon$, то значение свойства не определено);
- $Cur = \{cur\_ctx\}_{1:p}$ — множество значений свойств контекста на текущий момент времени $\forall j \in 1:p \quad cur\_ctx_j \in U$;
- $calc(x)$ для некоторого $x \in P \cup C$ — функция подсчета свойства сообщения (свойства контекста) x, возвращает текущее значение свойства.

Определим следующие отношения:

- $Dep\_prop(x, y)$ — отношение зависимости свойств. Значение свойства сообщения зависит от других свойств и свойств контекста, для которых отношение выполнено

$$Dep\_prop : (P \setminus A) \times (P \cup C) \to \{true, false\};$$

- $Dep\_Ctx(x, y)$ отношение зависимости свойств. Значение свойства контекста зависит от свойств сообщения и других свойств контекста, для которых отношение выполнено

$$Dep\_Ctx : C \times (P \cup C) \to \{true, false\}.$$

Данные отношения определяются функцией calc следующим образом:

- Если в функцию $calc(x)$ входит $calc(y)$, $y \in P$, то
  - если $x \in P \implies Dep\_prop(x, y) = true$;
  - если $x \in C \implies Dep\_ctx(x, y) = true$.
- Если в функцию $calc(x)$ входит cur_ctx_z, $z \in C$, то
  - если $x \in P \implies Dep\_prop(x, z) = true$;
  - если $x \in C \implies Dep\_ctx(x, z) = true$.

Определим операции данного аппарата:

- $get(x)$ — функция получения текущего значения свойства x;

- $get(z)$ — функция получения текущего значения свойства контекста z;
- $init\_ctx(z, init\_val)$ — функция инициализации значения свойства контекста z;
- $update\_ctx(z)$ — функция обновления значения свойства контекста z.

4. Обобщенный алгоритм восстановления порядка обмена сообщениями

Определим триггер цикла обработки *xchg_trigger*, как некоторое логическое выражение, при выполнении которого необходимо обновить свойства контекста. Как правило, этот триггер включается при поступлении нового сетевого пакета в анализатор. Обобщенный алгоритм восстановления порядка обмена сообщениями может быть формально описан как

$$A(C', NC', Dep\_ctx, xchg\_trigger, \{init\_val\}_{NC'}, \{calc\}_{C'}),$$

где C' — подмножество свойств контекста C, NC — пересечение C' и NC для пакета, характеризующегося множеством $(P, Dep\_prop, \{calc\}_P)$

```
begin
    foreach z ∈ NC   init_ctx(z, init_val_z)
    while xchg_trigger
        foreach z ∈ C   update_ctx(z)
    end while
end
```

Докажем следующее

Утверждение. *Обобщенный алгоритм восстановления порядка обмена сообщениями на базе описанного формального представления сетевых протоколов обладает, по крайней мере, такой же выразительной мощностью, какой обладает линейно-ограниченный автомат (машина Тьюринга на ленте ограниченной длины).*

Доказательство. Пусть линейно-ограниченный автомат, работающий на конечной ленте длины m, определен следующим образом:

- $S = s_{1:n}$ — множество состояний;
- $F \subset S$ — множество конечных состояний;
- $I = i_{1:p}$ — множество печатных символов;

- $N\colon S \times I \to S$ — функция перехода из состояния в состояние;
- $O\colon S \times I \to I$ — функция вывода на ленту;
- $D\colon S \times I \to \{-1;\ 0;\ +1\}$ — функция перемещения по ленте.

Не ограничивая общности, перенумеруем состояния от 1 до n, печатные символы от 1 до p. Ячейки ленты перенумеруем слева направо от 1 до m. Пусть содержащаяся на ленте программа составлена последовательностью символов $\{sym_1, \ldots, sym_m\}$.

Множество свойств пакета P имеет мощность 3:

- $P = \{prop_1, prop_2, prop_3\}$;
- $V_1 = S = \{s\}_(1 : n)$;
- $V_2 = I = \{i\}_(1 : p)$;
- $V_3 = \{-1;\ 0;\ +1\}$.

Множество свойств контекста C имеет мощность $m + 3$:

- $C = \{ctx_1, ctx_2, \ldots, ctx_3\}$;
- $U_1 = S = \{s\}_(1 : n)$;
- $U_2 = \{1, \ldots, m\}$;
- $U_3 = I = \{i\}_(1 : m)$;
- $U_4 = \ldots = U_(m + 3) = I = \{i\}_(1 : p)$.

Определим функции *calc* следующим образом:

- $calc(prop_1) = N(cur\_ctx_1, cur\_ctx_3)$ (следующее состояние);
- $calc(prop_2) = O(cur\_ctx_1, cur\_ctx_3)$ (выводимый символ);
- $calc(prop_3) = D(cur\_ctx_1, cur\_ctx_3))$ (перемещение);
- $calc(ctx_1) = calc(prop_1)$ (текущее состояние);
- $calc(ctx_2) = cur\_ctx_2 + calc(prop_3)$, функция r задается табличным способом (текущий номер ячейки);
- $calc(ctx_3) = r(cur\_ctx_2)$ (текущий символ);
- $calc(ctx_(i + 3))) = (cur\_ctx_2 = i)?(calc(prop_2)) : cur\_ctx_(i + 3), \forall i \in 1 : n$ (текущее содержимое ленты);

Примечание: функцию $calc(ctx_2)$ можно определить более аккуратно, с учетом ограниченности ленты в обоих направлениях (если выходит влево

или вправо за границы ленты, что соответствует -1 или $m+1$, то оставлять значение равным 0 или m соответственно).

Функция r: 1, 2, …, $m \to I$:

X	$r(x)$
1	cur_ctx_4
2	cur_ctx_5
…	…
M	cur_ctx_{m+3}

Триггер обработки можно определить следующим образом:

$$xchg_trigger() : ctx_1 \notin F.$$

Отношения зависимости между свойствами:

Dep_prop	$prop_1$	$prop_2$	$prop_3$	ctx_1	ctx_2	ctx_3	ctx_4	…	ctx_{m+3}
$prop_1$	false	false	false	true	false	true	false	false	false
$prop_2$	false	false	false	true	false	true	false	false	false
$prop_3$	false	false	false	true	false	true	false	false	false

Dep_ctx	$prop_1$	$prop_2$	$prop_3$	ctx_1	ctx_2	ctx_3	ctx_4	…	ctx_{m+3}
ctx_1	true	false	false	false	false	false	false	false	false
ctx_2	false	false	true	false	true	false	false	false	false
ctx_3	false	false	false	false	true	false	true	true	true
ctx_4	false	true	false	false	false	false	true	false	false
…	false	true	false	false	false	false	false	…	false
$ctx_m + 3$	false	true	false	true	false	true	false	false	true

Данные отношения зависимости формируют множество переменных контекста, подлежащих обязательной инициализации: $NC = ctx_1$, ctx_2, ctx_4, …, $ctx_(m+3)$, и последовательность операций для каждого шага эмуляции работы линейно-ограниченного автомата.

begin
$init\_ctx(cur\_ctx_1, s_0)$ (начальное состояние)
$init\_ctx(cur\_ctx_2, 1)$ (первая ячейка)
$init\_ctx(cur\_ctx_3, sym_1)$ (программа для линейно-ограниченного автомата)
…
$init\_ctx(cur\_ctx_(m+3), sym_m)$
while xchg_trigger()
$update(ctx_3)$ (считать текущий символ с ленты)
$update(ctx_4) \ldots update(ctx_(m+3))$ (вывести новый символ на ленту)

update(ctx_2) (переместить головку по ленте)
update(ctx_1) (перейти в новое состояние)
end while
end

Перенумеровав свойства контекста в соответствии с порядком обновления их в цикле работы, получим полную эмуляцию работы линейно-ограниченного автомата в описанном формальном представлении, что и требовалось доказать. □

Таким образом, доказано, что обобщенный алгоритм восстановления порядка обмена сообщениями применим для восстановления порядка работы любых останавливающихся алгоритмов обмена сообщениями.

Выразительность этого обобщенного алгоритма не превосходит выразительности линейно-ограниченного автомата при наложении на соотношения свойств контекста естественных ограничений ацикличности.

5. Применение обобщенного алгоритма восстановления порядка обмена данными в общем случае

Описанный в предыдущем разделе обобщенный алгоритм действует в линейном пространстве контекста, то есть когда все свойства контекста представляют собой равноправные элементы множества. Однако во многих случаях значения одних свойств контекста могут быть установлены только при определении значения других свойств контекста. Также может быть значимым порядок обработки свойств контекста. Например, если протокол реализует мультиплексирование/демультиплексирование виртуальных соединений, то длина уже переданных и полученных в рамках одного соединения данных может быть определена только после определения номера виртуального соединения. Кроме того, при неограниченном числе виртуальных соединений в линейном пространстве контекста невозможно установить значение свойства контекста, соответствующего длине переданных данных (множество значений не определено). Выходом является использование контекста с иерархической организацией. На вершине иерархии — множество свойств контекста, значения которых действительны для протокола в целом (версия, длина пакета, флаги приоритетов, номер виртуального соединения). На основании конкретных значений некоторых свойств этого множества и/или свойств пакета определяется множество значений свойств, действительных для отдельного виртуального соединения. Принятие решения о соответствии пакета конкретному экземпляру множества значений свойств виртуального соединения производится предикатом на

множестве значений свойств контекста, действительных для протокола, и/или свойств текущего пакета. Таким образом, получаем иерархическую (из двух уровней) древовидную структуру контекста: на первом этапе производится обработка множества свойств в корне дерева, на втором — принятие решения о выборе множества свойств контекста в одном из листьев и обработка этого множества. Обработка в каждом случае производится согласно обобщенному алгоритму.

Иерархическую организацию структуры контекста можно распространить на общий случай обработки сетевого пакета. Указанное разделение линейного пространства контекста на подмножества (далее называемые областями видимости) индуцируется характеристиками протокола обмена, связанными с выделением в этом протоколе транзакций, состоящих более чем из одного сообщения, а также инкапсуляцией протоколов. Таким образом, разделение областей видимости свойств контекста происходит при следующих обстоятельствах.

- Вследствие использования протоколом обмена данными механизмов мультиплексирования виртуальных соединений (пример — протокол TCP), выделения в рамках алгоритма обмена управляющих транзакций или транзакций обмена данными, состоящих более чем из одного сообщения (почтовые протоколы, протоколы передачи файлов) и тому подобных причин.
- Вследствие инкапсуляции протоколов. Протокол более низкого уровня имеет некоторый контекст, отдельные свойства которого должны быть видимы протоколу, который инкапсулируется в него. С другой стороны, протокол низкого уровня, за редким исключением, не имеет доступа к свойствам и к контексту вложенного протокола.
- В некоторых случаях, когда протокол описывает функции, обычно реализуемые на нескольких уровнях стека протоколов. Например, протокол прикладного уровня на базе UDP своими средствами реализует установление соединения, авторизацию и надежную доставку сообщений. Такая «эмуляция стека» также может быть описана с использованием разделения областей видимости контекста, также как в случае обычного разделения функций между службами различных уровней с использованием инкапсуляции.

Таким образом, контекст протокола представлен многоуровневой вложенной структурой, к которой рекурсивным образом применяется описанный в предыдущем разделе обобщенный алгоритм восстановления порядка обмена сообщениями.

6. Заключение

Описан подход к созданию формального декларативного представления сетевого протокола и обобщенного алгоритма для восстановления последовательности обмена сообщениями по некоторому сетевому протоколу, основанного на этом представлении. Показано, что полученный формальный аппарат применим для описания произвольного останавливающегося протокола обмена сообщениями.

Применение такого подхода облегчает труд программиста при создании средств разбора сетевых протоколов, позволяет сделать описание протоколов более прозрачным, упрощает верификацию этого описания. В свою очередь, повышается время разработки и надежность тех средств обеспечения сетевой безопасности, которым в работе необходимо производить разбор и восстановление порядка обмена сообщениями по различным сетевым протоколам.

Литература

[1] *Хопкрофт Д.*, *Мотвани Р.*, *Ульман Д.* Введение в теорию автоматов, языков и вычислений. М., 2002.

[2] `http://www.ietf.org/`.

Защита от сетевых атак методами фильтрации и нормализации протоколов транспортного и сетевого уровня стека TCP/IP

А. А. Чечулин, И. В. Котенко

1. Введение

В настоящее время неоспоримым фактом в области сетевой безопасности является огромный ущерб, наносимый сетевыми атаками. Существующие средства противодействия им не всегда справляются с новыми видами атак, поэтому актуальна задача создания такой системы, которая способна защитить не от конкретных атак, а от классов атак. Целью данной работы является: классификация атак, основанных на использовании стека протоколов TCP/IP; разбор и сравнение методов фильтрации и нормализации для каждого класса атак; разработка комплексного механизма фильтрации и нормализации трафика, предназначенного для использования на сетевом оборудовании.

Для аппаратной реализации методов на сетевом оборудовании методы должны обладать следующими свойствами: возможность обработки большого трафика за небольшое время (методы должны быть достаточно быстрыми и простыми); объем требуемой для работы памяти не должен превышать объем, доступный на оборудовании; методы обнаружения должны реализовать небольшое число обращений к памяти для обработки сетевых пакетов на высокой скорости.

Разработанный комплекс защиты может быть использован для защиты отдельного компьютера, защиты небольшой локальной сети (путем установки механизма на сетевом оборудовании), а также снижения нагрузки на межсетевые экраны в больших сетях.

Работа выполнена при финансовой поддержке РФФИ (проект № 07-01-00547), программы фундаментальных исследований ОНИТ РАН, Фонда содействия отечественной науке и при частичной финансовой поддержке, осуществляемой в рамках проекта Евросоюза RE-TRUST (контракт № 021186-2) и других проектов.

2. Виды фильтрации и нормализации трафика

Существующие сетевые атаки можно разделить на четыре класса:

- сбор информации, основанный на разнице в обработке корректных и некорректных пакетов;
- атаки, основанные на ошибках в обработке некорректных пакетов;
- сканирование хостов и сетей, основанное на использовании ошибок в обработке сессий;
- сканирование, основанное на корректном установлении соединений.

Для каждого класса атак предполагается использовать свой механизм защиты. Методы фильтрации и нормализации трафика можно разделить на два класса [1, 2, 3, 4]: методы, основанные на анализе отдельных пакетов, и методы, основанные на анализе последовательности пакетов.

Класс методов, основанных на анализе отдельных пакетов, состоит из следующих механизмов: механизмы фильтрации некорректных пакетов (обеспечивают защиту от атак, основанных на ошибках в обработке некорректных пакетов) и механизмы нормализации пакетов (обеспечивают защиту от сбора информации и сокрытия атак, основанных на разнице в обработке корректных).

Класс методов, основанных на анализе последовательности пакетов, состоит из следующих механизмов: механизмы фильтрации на основе данных о сессиях и семейство механизмов «Virus Throttling» (базовый и на основе метода CUSUM).

3. Фильтрация некорректных пакетов

Данный механизм предназначен для защиты от сбора информации и реализации атак с помощью ошибок в обработке некорректных сетевых пакетов. Для организации защиты используется фильтрация пакетов, имеющих некорректные заголовки. Примером работы механизма может служить защита от атаки Land — защита осуществляется с помощью фильтрации пакетов, у которых IP-адрес источника совпадает с IP-адресом получателя.

Достоинствами механизма основанного на фильтрации некорректных пакетов являются: уменьшение нагрузки на межсетевой экран; высокая скорость работы; надежность; возможность работы механизма на сетевом оборудовании; защита не от конкретных атак, а от класса атак, использующих ошибки в реализации протоколов.

Недостатками механизма являются: отсутствие дефрагментации пакетов; уязвимость к DoS-атакам (при слишком большом трафике пакеты либо задерживаются, либо пропускаются без обработки); пропуск атак, основанных на корректном сетевом взаимодействии.

4. Нормализация пакетов

Данный механизм служит для защиты от сбора информации и сокрытия атак с помощью разницы в реализации обработки сетевых пакетов. Для организации защиты используется нормализация пакетов, а именно — приведение полей заголовков пакетов к стандартному виду. Примером работы механизма может служить защита от сканирования топологии сети с помощью утилиты tracert — защита осуществляется с помощью изменения значения поля TTL протокола IP у всех входящих в сеть пакетов на 128.

Достоинствами механизма основанного на нормализации пакетов являются: уменьшение нагрузки на межсетевой экран; высокая скорость работы реализации; надежность, возможность работы механизма на сетевом оборудовании; защита не от конкретных атак, а от класса атак использующих разницу в реализации обработки протоколов.

Недостатками механизма являются: отсутствие дефрагментации пакетов; уязвимость к DoS-атакам (при слишком большом трафике пакеты либо задерживаются, либо пропускаются без обработки); пропуск атак, не использующих особенности обработки сетевых пакетов.

5. Фильтрация на основе данных о сессиях

Данный механизм предназначен для защиты от сбора информации и сокрытия атак с помощью TCP-пакетов, относящихся к несуществующим сессиям. Для организации защиты используется фильтрация пакетов, не имеющих отношения к корректно созданным сессиям и не являющихся частью корректного установления соединения. Примером работы механизма может служить защита от скрытого сканирования Stealth FIN (при сканировании данным методом используется отправка пакетов TCP с установленным флагом FIN для получения списка закрытых портов). Защита осуществляется с помощью фильтрация пакетов, не принадлежащих корректно созданным сессиям.

Достоинствами механизма основанного на фильтрации пакетов по данным о сессиях являются: уменьшение нагрузки на межсетевой экран; высокая скорость работы реализации; надежность; возможность работы ме-

ханизма на сетевом оборудовании (в сетях с небольшой сетевой активностью); защита не от конкретных атак, а от класса атак использующих ошибки в реализации работы с сессиями.

Недостатками механизма являются: сложность установки механизма на сетевом оборудовании в сетях с высокой сетевой активностью; уязвимость к DoS-атакам (при слишком большом трафике пакеты либо задерживаются, либо пропускаются без обработки); пропуск атак, использующих корректное создание сессий.

6. Фильтрация на основе сбора статистики

Методика «Virus throttling» («дросселирование/регулирование вирусов»), предложенная Вильямсоном, основывается на том факте, что легитимное приложение обычно демонстрирует стабильное число соединений с ограниченным числом внешних узлов.

В работе проанализировано несколько механизмов защиты, базирующихся на методике «Virus throttling» [5]:

- «virus throttling» для реализации на коммутаторе;
- «virus throttling» для реализации на коммутаторе на основе метода CUSUM.

Достоинствами механизмов основанных на методах «Virus Throttling» являются: простота реализации; эффективное обнаружение быстрого сканирования при условии «медленных» легитимных приложений; адаптивность — в процессе выполнения методики происходит регистрация наиболее часто используемых адресов.

Недостатками механизмов являются: блокировка хостов, на которых установлены приложения, генерирующие много запросов на соединение (web-браузеры, менеджеры закачки, P2P, прокси-сервера); невозможность обнаружения медленного сканирования; невозможность обнаружения сканирования, основанного на протоколе UDP; отсутствие обработки результата установления соединений.

В работе предполагается выполнить следующие улучшения механизмов:

- реализация анализа протокола UDP, представляя отдельный пакет как соединение;
- обработка ответов на запросы (TCP SYN-ACK), например, для игнорирования успешных соединений;

- использование для анализа не только IP-адресов, но других полей пакета, например, портов (для обнаружения сканирования портов);
- применение правил ACL (access control list) для игнорирования некоторых хостов (прокси-серверов и т.д.) или протоколов (HTTP и пр.);
- обработка сообщений от «honeypot» (если установлены в сети) и последующая блокировка адресов, обращающихся к «honeypot».

Заключение

В работе предложен комплексный подход к защите от сетевых атак, базирующийся на использовании механизмов нормализации и фильтрации сетевых пакетов.

На основе проведения имитационных экспериментов на разработанном программном средстве моделирования проанализированы механизмы защиты от сетевых атак, основанных на протоколах транспортного и сетевого уровней модели OSI, в том числе методики «Virus throttling» («virus throttling» для реализации на коммутаторе и «virus throttling» для реализации на коммутаторе на основе метода CUSUM), методики фильтрации трафика на основе данных о сессиях, методики нормализации и фильтрации некорректных пакетов и другие. Выявлены основные достоинства и недостатки предложенных механизмов защиты. Проведен ряд экспериментов.

Планируется проведение большой серии исследований на основе моделирования различных сетевых атак и предлагаемых механизмов защиты от них. Предполагается также осуществить следующие улучшения механизмов:

- оптимизация механизмов для возможности их работы на сетевом оборудовании;
- создание механизмов фильтрации для работы в сетях с P2P-трафиком;
- разработка системы фильтрации и нормализации протоколов верхнего уровня стека TCP/IP.

Литература

[1] *Handley M.*, *Paxson V.* Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics // AT&T Center for Internet research at ICSI (ACIRI International Computer Science Institute.

[2] *Lemonnier E.* Protocol Anomaly Detection in Network-based IDSs // Defcom 28th June 2001.

[3] *Mahoney M. V.* Network Traffic Anomaly Detection Based on Packet Bytes // Department of Computer Sciences Florida Institute of Technology.

[4] *Mahoney M. V.*, *Chan P. K.* PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic // Department of Computer Sciences Florida Institute of Technology.

[5] *Котенко И. В.*, *Чечулин А. А.* Исследование механизмов защиты от сетевых червей на основе методик Virus Throttling // Защита информации. Инсайд, № 3. 2008.

Антивирусная защита операционных систем штатными средствами

В. Г. Проскурин

В настоящее время в области обеспечения антивирусной защиты компьютеров и сетей связи доминирует подход, основанный на применении специализированных антивирусных программных или программно-аппаратных средств. При таком подходе на основе некоторых, как правило, субъективных предпочтений выбирается некоторый антивирусный комплекс, котрый устанавливается на все компьютеры защищаемой сети. Выполняется минимальный объем работ по его настройке и на этом все мероприятия по проектированию антивирусной защиты заканчиваются.

Данному подходу свойственны многочисленные недостатки, в том числе перечисленные далее:

- Ни один антивирусный сканер не обнаруживает абсолютно все вирусы. Если бы такой сканер существовал, его база сигнатур была бы неоправданно велика, а сканирование происходило бы недопустимо медленно.

- Антивирусное сканирование и контроль целостности программного обеспечения в большинстве случаев потребляют очень много вычислительных ресурсов компьютера. Как следствие — после установки антивирусного комплекса производительность системы существенно снижается.

- Сканирование, основанное на эвристиках атак, позволяет обнаруживать лишь те вирусы, в которых используются стандартные приемы реализации вирусом вредоносных функций. Нетрадиционные процедуры реализации вируса или грамотно проведенная обфускация его кода делают эвристический сканер абсолютно бесполезным против такого вируса.

- Эвристические сканеры, системы контроля целостности и антивирусные мониторы, как правило, генерируют неоправданно много ложных тревог.

- Контроль целостности программных файлов затрудняет установку нового и обновление ранее установленного программного обеспечения.
- Любой антивирусный монитор легко обходится вирусом, получившим доступ к ядру операционной системы.
- Антивирусные мониторы часто конфликтуют с другими видами программного обеспечения.
- Эксплуатация антивирусного комплекса требует от администратора системы высокой квалификации и значительных затрат. Как минимум, он должен регулярно просматривать журналы событий, зарегистрированных антивирусным комплексом, и следить за регулярым обновлением баз сигнатур известных вирусов.

С учетом перечисленных выше недостатков реальная эффективность современных антивирусных комплексов никогда не достигает заявленных значений. Для подтверждения данного тезиса был проведен эксперимент. Из сети Интернет было получено 50 образцов вредоносного машинного кода, из которых 15 образцов — в виде бинарных программных модулей и 35 образцов в виде исходных текстов, из которых были собраны бинарные программные модули, работоспособность которых была экспермиентально проверена. Все образцы вредоносного кода были установлены на виртуальную машину, изолированную от сети Интернет во избежание несанкционированного распространения исследуемых вирусов на другие компьютеры. На данную машину поочередно устанавливались следующие антивирусные комплексы: Avast, AVG, Avira, ClamWin, DrWeb, ESET NOD32, Norman Malware Cleaner, Panda, Антивирус Касперского. Все антивирусные комплексы, участвовавшие в эксперименте, были получены с сайтов разработчиков (пробные версии), перед началом эксперимента для каждого антивирусного комплекса было проведено обновление.

Суть эксперимента заключалась в сканировании изучаемыми антивирусными комплексами файловой системы виртуальной машины. Оценивалось количество образцов вредоносного кода, обнаруженных каждым антивирусным комплексом, отдельно для сигнатурного поиска (образцы вредоносного кода, полученные из Интеренет в виде бинарных программных модулей), и для эвристического поиска и антивирусного мониторинга (образцы вредоносного кода, собранные из исходных текстов непосредственно перед экспериментом).

Результаты эксперимента представлены на рис. 1 и 2.

Нетрудно убедиться, что современные антивирусные комплексы обеспечивают приемлемый для большинства пользователей уровень защищен-

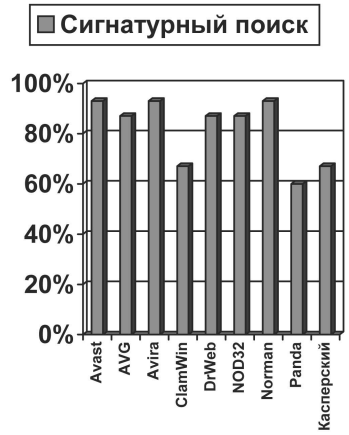


Рис. 1. Доля образцов вредоносного кода, обнаруженных исследуемыми антивирусными комплексами в результате эксперимента

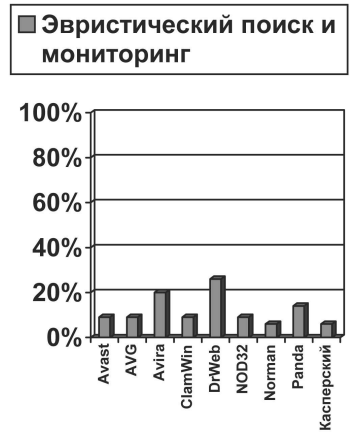


Рис. 2. Доля образцов вредоносного кода, обнаруживаемых заданным числом антивирусов с помощью эвристического поиска и мониторинга

ности лишь в отношении тех образцов вредоносного кода, которые присутствуют в его базе сигнатур. Если же вредоносная программа отсутствует в базе сигнатур антивирусного комплекса, вероятность обнаружения им данной программы не превышает 26%, При этом более половины таких программ не обнаруживаются ни одним из антивирусных комплексов, участвовавших в эксперименте.

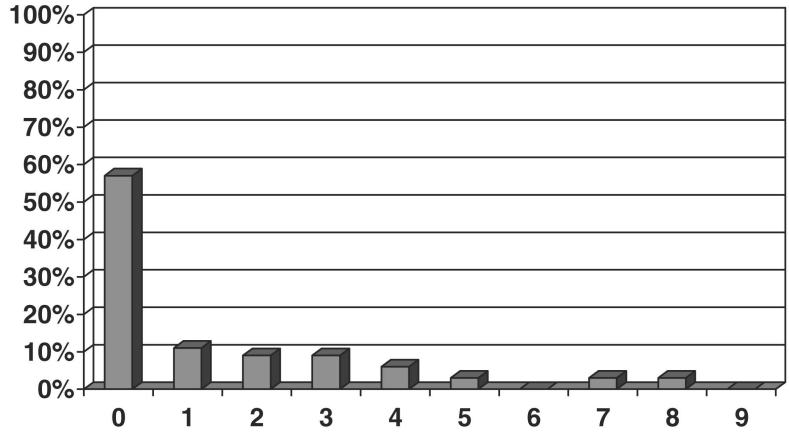


Рис. 3

В связи с изложенным представляет интерес вопрос об альтернативных подходах к обеспечению антивирусной защиты, менее обременительных для пользователей защищаемой системы. При этом будем принимать во внимание, что некоторое снижение устойчивости антивирусной защиты в большинстве случаев является допустимым.

Рассмотрим один из таких подходов, основанный на следующих мерах, которые следует трактовать, как меры обеспечения антивирусной защиты.

1. Реализация в защищаемой системе принципа минимизации полномочий пользователей. Каждому пользователю предоставляется набор полномочий, минимально достаточный для работы в системе. Учетные записи с административными полномочиями используются минимально, лишь тогда, когда это безусловно необходимо. Доступ администраторов к вирусоопасным объектам и средам (веб-браузеры, почтовые клиенты, программные файлы, установленные обычными пользователями) ограничен средствами разграничения доступа, встроенными в операционную систему. Благодаря перечисленным мерам снижается вероятность того, что вирус, проникший в систему, получит административные полномочия.

2. Реализация в защищаемой системе принципа минимизации её функциональных возможностей. В системе должен присутствовать лишь набор программного обеспечения, минимально необходимый для её работы по назначению и для работы пользователя. Чем меньше в системе установлено программного обеспечения, тем

меньше в системе будет уязвимостей, которые могут быть использованы вирусами.

3. Использование нетипичного программного обеспечения. Большинство современных вирусов ориентированы на поражение наиболее распространенного программного обеспечения (Microsoft Office, Internet Explorer, Outlook Express), программное обеспечение «второго эшелона» (Open Office, Opera, The Bat!,) уязвимо для вирусных атак в гораздо меньшей степени. Заметим, что этот факт связан не с плохой защищенностью популярного программного обеспечения, а с меньшей заинтересованностью вирусописателей в эксплуатации уязвимостей непопулярных программ.

4. Своевременная установка пакетов обновления. Начиная с 2003 года отмечена лишь одна уязвимость Windows [4], для которой использующий ее вирус появился раньше, чем устраняющий ее пакет обновления. Для других программных продуктов ситуация несколько хуже.

5. Применение пакетных фильтров. Пакетные фильтры практически не препятствуют непосредственно поражению защищаемой системы вирусами, однако существенно затрудняют дальнейшее распространение вируса из пораженной системы.

6. Организационные меры антивирусной защиты, включающие инструктирование пользователей и другие подобные им.

Перечисленные выше меры защиты могут показаться очевидными. Однако большинство пользователей в настоящее время полностью или частично пренебрегает ими, предпочитая подход, основанный на бездумном применении антивирусных программных комплексов. Определенную роль в этом играет активная реклама коммерческих антивирусов.

Начиная с 2000 года, в операционных системах семейства Windows начали появляться средства, позволяющие реализовать принцип минимизации полномочий не только на уровне пользователей, но и на уровне отдельных программ. К этим средствам относятся ограниченные маркеры доступа (Windows 2000), необратимое удаление привилегий из маркера доступа (Windows 2003), контроль учетных записей и уровни целостности (Windows Vista) [3]. Начиная с Windows Vista, такие средства позволяют обеспечить необходимый уровень антивирусной защищенности даже без искусственного ограничения полномочий администраторов. К сожалению, крайне низкие эксплуатационные качества Windows Vista ограничивают возможности применения этой операционной системы.

Политика безопасности, реализованная перечисленными мерами, не предусматривает автоматического выявления и блокирования атакующих систему вирусов. Однако каждый проникший в систему вирус обладает минимальными полномочиями, не позволяющими ему причинять серьезный вред системе и пользователю. Как правило, вирус при этом переходит в латентное состояние и никак не проявляет себя в течение длительного времени. Автору приходилось обнаруживать в исправно функционирующих экземплярах Windows XP до пятнадцати различных вирусов в латентном состоянии. Удаление латентных вирусов реализуется с помощью антивирусного сканера, запускаемого один-два раза в год либо при обнаружении проблем, вызванных некорректным функционированием вирусов в латентном состоянии.

Для дополнительного подтверждения высокой эффективности предлагаемого подхода был проведен эксперимент. В ходе эксперимента 50 вредоносных программ, участвоваших в предыдущем эксперименте, были запущены на выполнение с полномочиями обычного пользователя в политике безопасности, принятой в Windows XP по умолчанию. Лишь одна программа смогла выполнить вредоносные действия, 46 программ запустились, однако перешли в латентное состояние и 5 вредоносных программ аварийно завершились немедленно после запуска.

Важным достоинством предлагаемого подхода является тот факт, что он не предполагает закупок дополнительного программного обеспечения (ежегодное сканирование системы может быть выполнено любым бесплатным антивирусным сканером, например, CureIt). Кроме того, нет необходимости в расходах на трафик обновления антивирусных баз, исключаются конфликты резидентных антивирусных сканеров и мониторов с другими программами.

Единственным серьезным недостатком рассматриваемого подхода является возможность фатального поражения системы при одновременном выполнении следующих условий:

- в защищаемой системе имеется уязвимость, позволяющая вирусу несанкционированно получить администинистративные полномочия;
- пакет обновления, устраняющий данную уязвимость, по каким-то причинам не был установлен;
- вирус, проникший в систему, является опасным или особо опасным.

Следует заметить, что опыт практической эксплуатации защищенных компьютерных систем подтверждает, что при точном соблюдении принципа минимизации полномочий данная ситуация не фиксировалась ни разу. Известные случаи серьезного повреждения системы вирусом всегда были

связаны с нарушениями администраторами данного принципа. Необходимо также отметить, что в рамках формальных моделей систем управления доступом и информационными потоками (расширенная модель Take-Grant [2], ДП-модели [1]) возможно формальное доказательство невозможности нанесения вреда доверенным субъектам системы со стороны недоверенного (непривилегированного) субъекта-вируса.

Литература

[1] *Девянин П. Н.* Анализ безопасности управления доступом и информационными потоками в компьютерных системах. М.: Радио и связь, 2006, 176 с.

[2] *Frank J.*, *Bishop M.* Extending the Take-Grant Protection System. Department of Computer Science. University of California at Davis, 1984.

[3] About Authorization. `http://msdn.microsoft.com/en-us/library/aa374702(VS.85).aspx`

[4] Microsoft Security Bulletin MS06-001. `http://www.microsoft.com/technet/security/bulletin/MS06-001.mspx`

К вопросу имитационного моделирования механизмов разделения коммуникационных ресурсов компьютерных сетей

В. Б. Савкин

1. Введение

Традиционным и, как правило, применяемым принципом доставки пакетов в IP-сетях является принцип наилучших усилий (best effort), согласно которому все пакеты находятся в равных условиях. Использование такого принципа обслуживания означает, что один пользователь или одно приложение может загрузить канал, а негативные последствия (рост задержек, увеличение вероятности потерь пакетов) ощущают все пользователи и все приложения в равной степени. Отмеченные последствия имеют место и в том случае, когда выделяются специальные приложения, для которых осуществляется приоритетное обслуживание, либо когда трафик делится на классы приоритетности. В этом случае также необходимо пропускную способность каналов каким-то образом разделить между всеми потребителями, имеющими одинаковый приоритет. С тем, чтобы предотвратить эти негативные эффекты для всех пользователей, поставщики услуг связи вынуждены использовать каналы в таком режиме, когда средняя загрузка очень далека от максимальной. Это может достигаться, например, ограничением полосы, доступной пользователям (в частности, подключением пользователей при помощи медленных линий связи), или путём использования экономических рычагов (оплата за услуги связи в зависимости от объема переданных данных).

Таким образом, можно поставить задачу справедливого разделения ресурса, а именно, пропускной способности канала, между потребителями. Под справедливостью в контексте данной публикации понимается как можно меньшая зависимость доли ресурсов, доступных одному пользователю или приложению, от поведения остальных пользователей или приложений. Следует также учитывать гранулярность, или точность классификации пакетов по их потребителю. Идеальной, по видимому, является

ситуация, когда каждый пакет соотносится с индивидуальным конечным пользователем и с одной из его задач. В этом случае можно рассчитывать на нормальную работу широкого класса приложений при практически полностью загруженном канале. Такой вывод следует из того факта, что в сети Интернет большую часть пропускной способности каналов связи потребляют приложения, которые наименее чувствительны к временному падению скорости передачи в «часы пик». В первую очередь к приложениям с данным свойством относятся различные службы передачи файлов. Внедрение механизмов справедливого разделения ресурсов сможет ограничить полосу, выделяемую таким «жадным» приложениям, не снижая качество работы интерактивных приложений, тогда как при отсутствии подобных механизмов интерактивные приложения страдают. Настоящая публикация подтвержает данные рассуждения результатами имитационного моделирования.

2. Постановка задачи справедливого и эффективного распределения ресурсов

Для задачи справедливого и эффективного распределения ресурсов интерес представляют такие показатели, как зависимость доли ресурсов, доставшихся данному потоку, от поведения других потоков (показатель справедливости) и средняя загрузка каналов (показатель эффективности). Более детально, справедливое разделение пропускной способности означает выполнение следующих условий.

1. Существует гарантированная минимальная доля пропускной способности, на которую может претендовать поток. Этот факт даёт возможность утверждать (при достаточной интенсивности потока на входе в сеть), что заранее определенный объем данных успешно пройдет через сеть и будет получен на выходе. Интерес также представляют такие параметры, как максимальные и средние задержки, при условии выполнения ограничений, подобных введенным выше для задач реального времени.
2. Существует алгоритм разделения неиспользованной потоком доли между другими потоками, например, пропорционально некоторым весам, определенным для всех потоков.

Задача разделения полосы пропускания одного канала между конечным числом потоков, связаннных с пользователями или приложениями, чьи требования заранее известны и перечислены, является довольно хорошо исследованной. Для её решения разработаны специальные дисциплины

очередей, а именно Class-Based Queueing. Алгоритмы для таких очередей предложены, например, в работах [1] и [2], получены оценки для задержек, создаваемых очередью, например, в работе [3]. Существуют как свободные программные реализации подобных алгоритмов, так и реализации в некоторых моделях сетевого оборудования ведущих производителей. Однако, в данной работе рассматривается ситуация, когда приложения запускаются и завершаются пользователями динамически, и требования каждого из них администраторам сети не известно.

Задачу эффективного использования пропускной способности каналов можно сформулировать следующим образом: какие значения загрузки каналов достижимы при условии сохранения ожидаемого пользователями качества обслуживания? Получение точного ответа на данный вопрос затруднительно в свете того, что для этого потребуется некоторая модель поведения и ожиданий пользователей. По этой причине часто ограничиваются эмпирическими оценками. Считается, например, что (без использования специальных методов справедливого распределения) многодневное среднее от загрузки канала не должно превышать 30%–50% от пропускной способности, иначе в «часы пик» качество обслуживания недопустимо падает. Следует ожидать, что внедрение механизмов справедливого деления пропускной способности позволит заметно увеличить этот показатель при одновременном улучшении качества обслуживания, предоставляемого интерактивным приложениям (за счет уменьшения доли ресурсов, потребляемых «жадными» пользователями и приложениями). Для проверки данного тезиса автор использовал имитационное моделирование, результаты которого представлены в следующих подразделах.

1. Модель справедливой очереди

Для исследования поведения справедливой очереди был использован симулятор сетей ns-2 [4]. Автором реализован простой алгоритм справедливой очереди. Справедливая очередь состоит из подочередей типа FIFO, по одной на каждый поток. Алгоритм обходит подочереди по кругу (такой класс алгоритмов называется round-robin) и выбирает из каждой порцию данных, то есть несколько пакетов. Переход к следующей подочереди осуществляется в том случае, когда

- либо в текущей подочереди не осталось пакетов,
- либо из текущей подочереди в данном раунде было суммарно выбрано не менее q байт.

Кроме кванта данных q, к параметрам алгоритма относится ограничение на длину очереди, которое в данной реализации представляет максимальное

число пакетов P. При превышении этого ограничения алгоритм удаляет пакет из конца подочереди, содержащей в данный момент наибольшее число пакетов.

Алгоритм был реализован в виде класса на языке C++ с привязкой к языку Tcl, как это требуется для модулей системы ns-2.

Для очереди, работающей по данному алгоритму, получена следующая оценка доли r_g пропускной способности канала, на которую может рассчитывать [1] каждый поток

$$r_g \geqslant \frac{qS}{Nq + (N-1)M},$$

где S — пропускная способность канала, N — число активных потоков и M — максимальный размер пакета (MTU). Действительно, для любого выбранного потока за один раунд алгоритм вынимает из соответствующей подочереди не менее q данных (рассматривается случай, когда в этой подочереди всегда достаточно много пакетов), а данных из остальных подочередей — не более $(N-1)(q+M)$ в сумме.

Рассмотренный алгоритм очереди отличается простотой реализации за счёт точности разделения полосы. Однако, несмотря на эту особенность, он может продемонстрировать отличия справедливой очереди от обычного подхода best effort.

Цель проведённого автором имитационного моделирования состояла в сравнении работы справедивой очереди и обыкновенной очереди типа FIFO при условии одновременной активности «жадных» и интерактивных приложений. При этом предполагалось, что влияние интерактивных приложений на загрузку канала мало, и основная нагрузка создаётся «жадными» приложениями. В симуляторе была построена простейшая модель сети типа «бутылочное горлышко». Модель сети содержала 100 узлов-генераторов, имитирующих «жадные» приложение, два генератора, имитирующих передачу мультимедийного трафика, и один генератор интерактивного трафика, который моделировал процесс просмотра пользователем веб-страниц.

«Жадные» приложения моделировались как передача больших объёмов данных с использованием протокола TCP (вариант Reno). Нагрузка, создаваемая данными приложениями, описывается *показателем нагрузки*:

$$\gamma = \frac{N_l B}{ST_E} \cdot 100\%,$$

[1] При достаточной интенсивности потока на входе в очередь интенсивность потока на выходе, составленного из пакетов, которые не были отброшены очередью и были выбраны алгоритмом round-robin, составит не менее r_g.

где N_l — число запросов на передачу данных, инициированных в ходе эксперимента, B — объём данных, запланированных к передаче в ответ на каждый запрос[2], T_E — время эксперимента. Во всех проведённых экспериментах были выбраны значения $T_E = 14400$ с (4 часа) и $B = 10^{10}$ байт. Данный показатель игнорирует накладные затраты протоколов передачи файлов, но позволяет оценить «потребности пользователей» относительно возможностей канала. В проведённых экспериментах показатель нагрузки менялся от низкого уровня (40%) до состояния крайней перегрузки ($>> 400\%$).

На «бутылочное горлышко» ставилась как очередь типа drop-tail, так и справедливая очередь. Кроме того, были использованы две разных модели активности пользователей. В одной из них каждый пользователь мог запускать параллельно несколько TCP-соединений, а в другой пользователь ожидал окончания передачи одного файла, прежде чем запустить следующую. Таким образом, всего было проведено 4 иммитационных эксперимента для каждого значения показателя нагрузки.

В качестве выходных значений симуляций рассматривались показатели качества обслуживания для мультимедийных и гарантированных приложений. Мультимедийные приложения моделировались потоками UDP-пакетов постоянной интенсивности: 64 кбит/с для одного потока, что меньше гарантированной полосы для данных условий, и 256 кбит/с для другого потока. В качестве показателей качества обслуживания для данных потоков рассматривались доли пакетов, задержка которых при передаче от генератора к приёмнику превышала некоторый порог[3] Качество передачи для некоторого конкретного значения порога можно считать удовлетворительным, если не более 5% пакетов не уложились в данную величину задержки. Заметим, что такая доля не доставленных пакетов не приводит к ухудшению разборчивости речи при использовании технологии VoIP.

Процесс просмотра веб-страницы моделировался как установление TCP-соединения и передача 10000 байт данных. Качество обслуживания веб-запросов считалось удовлетворительным, если все передачи завершались не более чем за 30 с. Кроме данного критерия, вычислялись медиана и 90-й персентиль эмпирического распределения времени выполнения веб-запроса. Для вычисления всех отмеченных выше показателей по протоколам работы симулятора автором были написаны программы на языках Ocaml и Perl.

[2]Не все запланированные к передаче данные могут быть переданы в ходе эксперимента из-за ограничения на время.

[3]Отброшенные пакеты считаются имеющими бесконечно большую задержку.

3. Результаты моделирования

Результаты иммитационного моделирования показывают, что использованием справедливой очереди можно добиться достаточно хорошей работы интерактивных приложений даже в ситуации крайней перегрузки. Так, иммитируемые веб-запросы обрабатываются удовлетворительно независимо от показателя нагрузки при использовании справедливой очереди, и неудовлетворительно при использовании FIFO в условиях перегрузки.

Высокоскоростное мультимедийное приложение быстро достигает неудовлетворительных показателей при обоих типах очередей. Можно сделать вывод, что для работы подобных приложений в условиях перегрузки нужны другие средства управления качеством обслуживания, например, явное задание приоритета для выделенных приложений.

Показатели качества обслуживания низкоскоростных мультимедийных приложений при росте нагрузки плавно деградируют до определённого предела при использовании справедливой очереди и более резко достигают полностью неудовлетворительного уровня при использовании FIFO.

4. Выводы

Для обсуждения полученных результатов нужно рассмотреть рост потребностей пользователей со временем. На временных промежутках порядка нескольких лет он хорошо приближается экспонециальным законом [5]. Одновременная резкая деградация в обслуживании всех типов приложений означает внезапное ухудшение оценки уровня сервиса пользователями и, в условиях конкуренции, резкий отток клиентов. Постепенная деградация, начиная с самых требовательных приложений, например, высокоскоростных видеотрансляций, позволяет спланировать развитие сети с учётом необходимости расширения каналов связи, установки нового оборудования, внедрения новых механизмов управления трафиком. Данные соображения позволяют сделать вывод о наличии потенциального экономического эффекта от внедрения описанного вида механизмов разделения полосы пропускания. В литературе описаны дисциплины очередей с ещё более хорошими показателями, как в плане справедливости, так и в плане вычислительной эффективности и пригодности для их реализации «в железе» [6].

Данная работа предлагает альтернативу общепринятому на текущий момент подходу к развитию сетей путём опережающего расширения каналов с избеганием перегрузок. Такой подход представляется наиболее оправданным для сетей магистральных операторов связи, однако, по неко-

торым экспертным оценкам [5], он может стать неприменимым вследствие экономических факторов. Результаты представленного исследования показывают, что технология справедливого разделения полосы пропускания может сэкономить использование ресурсов и, как следствие, позволить более успешно развиваться сетям в условиях ограниченной возможности расширения каналов связи.

Литература

[1] *Floyd S.*, *Jacobson V.* Link-sharing and Resource Management Models for Packet Networks // IEEE/ACM Transactions on Networking. 1995. V. 3, №. 4.

[2] *Shreedhar M.*, *Varghese G.* Efficient fair queuing using deficit round-robin // IEEE Transactions on Networking. June 1996. V. 4, №. 3. P. 375–385.

[3] *Kanhere S. S.*, *Sethu H.* On the Latency Bound of Pre-Order Deficit Round Robin // Proceedings of the IEEE Conference on Local Computer Networks. November 2002.

[4] The Network Simulator — ns-2. `http://www.isi.edu/nsnam/ns/`.

[5] *Кипчатов А.* Рынок магистрального IP-транзита РФ до 2010 года: тенденции, емкость, цены. Доклад на Пиринговом форуме MSK-IX. 2007. `http://www.msk-ix.ru/download/forum2007/IPtransit_market.ppt`.

[6] *Kanhere S. S.*, *Sethu H.* Prioritized Elastic Round Robin: An Efficient and Low-Latency Packet Scheduler with Improved Fairness. Technical Report DU-CS-03-03, Department of Computer Science, Drexel University, Philadelphia, PA 19104, July 2003.

Методы и программное обеспечение решения задач управления безопасностью объектов транспортной инфраструктуры

А. А. Кононов

Федеральный Закон «О транспортной безопасности» [2] четко определяет те процедуры, которые должны быть решены в рамках автоматизированной системы обеспечения транспортной безопасности (АС ОБ). К таким процедурам в Законе отнесены:

- оценка уязвимости объектов транспортной инфраструктуры и транспортных средств от актов незаконного вмешательства (статья 5 Закона);
- категорирование объектов транспортной инфраструктуры и транспортных средств (статья 6 Закона);
- разработка и контроль выполнения требований по обеспечению транспортной безопасности (статья 8 Закона);
- планирование и реализация мер по обеспечению транспортной безопасности объектов транспортной инфраструктуры и транспортных средств (статья 9 Закона);
- информационное обеспечение в области транспортной безопасности (статья 11 Закона).

В соответствии с Концепцией [1] работа систем обеспечения безопасности сложных технических и организационных объектов должна строиться на основе *оценки рисков нарушения их безопасности*. Это положение было принято в качестве исходного при разработке АС ОБ дорожной службы. Аппаратно-программный комплекс «РискМенеджер», использованный для реализации автоматизированной системы, достаточно хорошо описан в целом ряде публикаций [3, 4, 5]. В данной статье обратим внимание на то, что в нем четко реализована вся процедура оценки рисков нарушения безопасности объектов любых видов транспорта, которых в контексте данной публикации будем именовать опасными объектами.

Рассмотрим методологию выполнения процедур обеспечения безопасности опасных объектов дорожного хозяйства (ДХ).

Процедура категорирования опасных объектов ДХ проводится с целью определения минимального числа категорий. В этом случае объектам каждой из этих категории должны быть сопоставлены единые требования по обеспечению безопасности, стандартная модель угроз, стандартный профиль защиты и рекомендуемая практика обеспечения безопасности.

Опасными объектами инфраструктуры ДХ являются искусственные сооружения на федеральных автомобильных дорогах — мосты, тоннели, путепроводы и противолавинные галереи. Главной конструктивной особенностью этих объектов является их относительно простая, линейная и практически однотипная структура. Этот факт означает, что все они имеют стандартный набор элементов, что позволяет построить для них стандартную модель угроз и стандартный профиль защиты. Например, любой мост имеет пролеты, опоры и так далее и, как правило, одинаковый набор уязвимых элементов. Такие характеристики относятся к тоннелям и к другим сооружениям ДХ. Исходя из конструктивных особенностей объектов ДХ для категорирования по степени потенциальной опасности их можно свести к двум типам сооружений — *мостовые сооружения и тоннельные сооружения*.

Основанием внутриотраслевой классификации этих двух типов объектов служит их основной параметр — длина. Так к первому классу относят мосты и тоннели, длина которых достигает 500 метров и более. Ко второму классу отнесены сооружения, длина которых составляет от 300 до 500 метров, к третьему классу — сооружения длиной от 100 до 300 метров и к четвертому классу сооружения длиной до 100 метров. В соответствии с этой классификацией число категорий опасных объектов ДХ целесообразно принять равным числу их классов, а именно — четырем.

Категорирование опасных объектов ДХ проводится по величине потенциального ущерба, который может быть получен в результате реализации деструктивного воздействия. Такое воздействие в настоящей статье будем именовать террористической атакой, имея в виду террористические мотивы её подготовки и проведения.

Категории опасных объектов определяются в соответствии с диапазонами ожидаемого потенциального ущерба, указанными в таблице 1.

Оценка потенциального ущерба в результате разрушения опасного объекта ДХ осуществляется по интегральному критерию $К^{r}_{инт}$:

$$К^{r}_{инт} = (К_{л} + К_{экон} + К_{бал} + К_{экол})Р_{с}, \qquad (1)$$

где

- $К_{л}$ — финансовый ущерб, определяемый численностью погибших и пострадавших, в случае реализации террористической атаки на объект;

Таблица 1. Диапазоны ущербов по категориям опасных объектов ДХ

КАТЕГОРИИ	ДИАПАЗОН УЩЕРБА
1-я категория	500 млн. руб. и выше
2-я категория	от 250 до 500 млн. руб.
3-я категория	от 75 до 250 млн. руб.
4-я категория	от 2,5 млн. до 75 млн. руб.

- $К_{экон}$ — финансовый ущерб от уменьшения грузопотока и пассажиропотока в результате вывода из строя наиболее уязвимых элементов объекта;
- $К_{бал}$ — балансовая стоимость сооружения (или стоимость восстановления);
- $К_{экол}$ — стоимостное выражение ожидаемого экологического ущерба в случае реализации атаки на объект;
- $Р_с$ — уровень террористической опасности в регионе.

Для каждого типа и класса объектов ДХ рассчитываются значение интегрального ущерба и, в зависимости от полученного результата, объекты относят к соответствующей категории.

Важной особенностью расчетов потенциального ущерба опасным объектам ДХ является высокая степень неопределенности исходной информации. Такая неопределенность прежде всего относится к параметрам дорожного движения на момент террористической атаки, к структуре и другим характеристикам грузопотока, к стоимостным оценкам потенциальных потерь грузов, последствий их опоздания и так далее. Эта неопределенность объективно присуща рассматриваемым процессам и не может быть разрешена какими-либо научными методами. Однако это обстоятельство не мешает построить систему огрубленных оценок потенциальных потерь, вполне достаточную для распределения опасных объектов по категориям опасности. При этом важна не столько точная оценка потенциального ущерба объекту, сколько сравнительная оценка этих ущербов для разных классов объектов, чтобы разнести их по категориям. Этому способствуют и достаточно широкие диапазоны потенциального ущерба для каждой категории объектов, представленные в таблице 1.

Чтобы это сравнение было корректным, необходимо чтобы условия расчетов для всех объектов были одинаковы. Такое возможно, если значительная часть параметров заранее задана в качестве принятого стандарта, например, средняя скорость грузопотока, средняя стоимость транспортного средства и груза. Для определения стандартных значений расчетных

параметров необходимо для каждого типа объектов построить некоторый наиболее опасный сценарий развития ситуации и выбирать расчетные параметры, обеспечивающие корректное сравнение объектов по потенциальному ущербу. Затем для каждого такого параметра определить некоторые средние значения, которые будут использоваться для всех расчетов.

Приведенная ниже исходная информация, необходимая для категорирования опасных объектов дорожного хозяйства, должна быть отражена в паспорте безопасности каждого объекта, который служит юридическим основанием для расчета его индекса.

Расчет интегрированного ущерба в случае реализации террористических атак на опасные объекты ДХ

Мостовые сооружения

1. Расчет интегрированного ущерба в случае террористической атаки на мостовые сооружения производится из предположении о максимальном ущербе сооружению — разрушении (обвале) одного или двух (в зависимости от типа конструкции) пролетов моста в результате подрыва его опор. Расчеты проводятся последовательно для всех составляющих $K^{r}_{инт}$ (см. формулу (1)).

Ущерб $K_{л}$, определяемый численностью погибших и пострадавших, в случае реализации террористической атаки на объект рассчитывается по среднему числу транспортных средств, которые могут находиться на обрушившихся пролетах сооружения в момент совершения террористической атаки:

$$K_{л} = \frac{(L_{пр} + L_{тп}) \bullet (G_{г} \cdot S_{г} + G_{п} \cdot S_{п})}{V} \cdot C_{л}, \qquad (2)$$

где

- $L_{пр}$ — суммарная длина упавших пролетов в метрах;
- $L_{тп}$ — длина тормозного пути в метрах в зависимости от скорости транспортного средства;
- $G_{г}$ — грузопоток — количество транспортных средств, проходящих какую либо точку трассы в минуту;
- $G_{п}$ — пассажиропоток — количество пассажирских транспортных средств, проходящих начальную линию мостового сооружения в минуту;
- $S_{г}$ — среднее количество людей в грузовых транспортных средствах;

- $S_{\text{п}}$ — среднее количество людей в пассажирских транспортных средствах;
- V — средняя скорость транспортных средств метров в мин.;
- $\text{С}_{\text{л}}$ — цена ущерба.

Ущерб $\text{К}_{\text{экон}}$ определяется как сумма стоимости выведенных из строя транспортных средств $\text{К}^{\text{тс}}_{\text{экон}}$, которые могут находиться на обрушившихся пролетах сооружения в момент совершения террористической атаки; см. рис. 1) и ущерба от уменьшения или прекращения грузопотока $\text{К}^{\text{г}}_{\text{экон}}$ в период от разрушения до восстановления моста:

$$\text{К}_{\text{экон}} = \text{К}^{\text{тс}}_{\text{экон}} + \text{К}^{\text{г}}_{\text{экон}}. \tag{3}$$

Ущерб $\text{К}^{\text{тс}}_{\text{экон}}$ рассчитывается по формуле

$$\text{К}^{\text{тс}}_{\text{экон}} = \frac{(L_{\text{пр}} + L_{\text{тп}}) \bullet G}{V} \cdot \text{С}_{\text{тс}}, \tag{4}$$

где

- G — суммарный грузопоток — количество транспортных средств, проходящих начальную линию мостового сооружения в минуту;
- $\text{С}_{\text{тс}}$ — средняя стоимость транспортного средства и груза в зависимости от структуры грузопотока.

Ущерб $\text{К}^{\text{г}}_{\text{экон}}$ рассчитывается для трех основных ситуаций, которые могут возникнуть при совершении террористической атаки на федеральных автомобильных дорогах.

1-*я ситуация*, когда на участке трассы, где разрушен мост, имеется объезд или возможно возведение понтонного моста. В случае, когда объезд небольшой, ущерб от уменьшения грузопотока рассчитывается по формуле

$$\text{К}^{\text{г}}_{\text{экон1}} = (G - G_{\text{об}}) T_{\text{вос}} \text{С}_{\text{г}} I_{\text{шт}}(T_{\text{вос}}). \tag{5}$$

Если объезд более 200 км, то учитывается опоздание груза за время объезда:

$$\text{К}^{\text{г}}_{\text{экон1}} = (G - G_{\text{об}}) T_{\text{вос}} \text{С}_{\text{г}} I_{\text{шт}}(T_{\text{вос}}) + T_{\text{об}} G_{\text{об}} I_{\text{шт}}(T_{\text{об}}), \tag{6}$$

где

- $G_{\text{об}}$ — пропускная способность (максимальный грузопоток) объездного пути или понтонного моста;
- $T_{\text{вос}}$ — время восстановления моста;
- $T_{\text{об}}$ — время (в часах) объезда или наведения понтонного моста:

$$T_{\text{об}} = \frac{L_{\text{об}}}{V_{\text{об}}}; \tag{7}$$

- $L_{об}$ — длина объездного пути;
- $C_г$ — средняя стоимость груза в зависимости от структуры грузопотока;
- $V_{об}$ — средняя скорость движения по объездному пути в км/час;
- $I_{шт}$ — средняя величина штрафа в процентах от стоимости груза в зависимости от времени опоздания.

Если суммарное время опоздания груза в результате разрушения моста менее суток, то ущерб $К^г_{экон1}$ *не принимается в расчет.*

Средняя величина штрафа за опоздание груза может колебаться в широких пределах от 10% до 50% и более стоимости груза в зависимости от его свойств и характера использования. Например, для скоропортящихся продуктов питания определены предельные сроки их доставки. В случае превышения этих сроков, штраф может составлять полную стоимость груза, к которой добавляется неустойка. Если груз предназначен для обеспечения непрерывного производства и его несвоевременная доставка может нарушить производственный процесс, то величина штрафа должна покрыть все связанные с опозданием издержки производителя.

Естественно, что точно учесть структуру и особенности грузов в грузопотоке чаще всего не представляется возможным. По этой причине для приближенных расчетов можно принять среднюю стоимость груза на одном транспортном средстве 500000 руб., а величину штрафа — 10% стоимости груза в сутки. В зависимости от известной структуры грузопотока на конкретном участке трассы эти цифры могут легко корректироваться.

2-*я ситуация*, когда объезда на трассе нет и транспорт вынужден дожидаться восстановления моста. В этом случае ущерб рассчитывается по формуле:

$$К_{экон} = GT_{вос}C_г I_{шт}. \tag{8}$$

3-*я ситуация*. Если мост является единственным путем жизнеобеспечения изолированных населенных пунктов и основу грузов составляют пищевые продукты, лекарства и другие необходимые для жизни товары, то стоимость ущерба определяется полной стоимостью не доставленных товаров умноженной на коэффициент ущерба жизни и здоровью людей — Y, значения которого могут составлять от 1, когда задержка составляет не более суток и до 10 и более, когда задержка составляет 10 и более суток. Коэффициент Y может быть уточнен для конкретной ситуации в каждом регионе. С учетом изложенного:

$$К_{экон} = G_г T_{вос}\, e\, C_г Y(T_{вос}) \tag{9}$$

где e — доля продуктов, необходимых для жизнеобеспечения населения изолированного населенного пункта.

2. Для упрощенной оценки $К_{бал}$ — стоимости восстановления моста можно воспользоваться его начальной балансовой стоимостью приведенной с помощью коэффициента перевода $F_{пер}$ к ценам текущего периода:

$$К_{бал} = С_{бал} F_{пер}. \tag{10}$$

Можно также воспользоваться средними оценками стоимости строительства моста в зависимости от его класса, которые закладываются в государственные проекты строительства мостов.

3. Оценка экологического ущерба $К_{экол}$, исходит из того, что экологический ущерб при разрушении мостов может быть нанесен транспортными средствами с опасными грузами, которые могут быть на мосту в момент его разрушения. Для расчетов необходимо знать долю опасных грузов в грузопотоке q_i, где i — тип опасного груза. Кроме того, необходимо знать нормативы ущерба в денежном выражении от аварии транспортного средства с каждым типом опасного груза — $С^i_{эк}$. Такие нормативы имеются в МЧС. Тогда ущерб $К_{экол}$ может быть рассчитан по формуле:

$$К_{экол} = \sum_{i=1}^{I} \frac{(L_{пр} + L_{тп}) \bullet G_{г}}{V} \cdot q_i С^i_{эк}, \tag{11}$$

где I — количество типов опасных грузов.

4. Суммарный ущерб, по которому производится категорирование опасных объектов ДХ, рассчитывается по формуле (2) с использованием таблицы 1.

Полученный результат сравнивается со значениями диапазонов ущербов в таблице 1 и объекту присваивается соответствующая категория.

Тоннельные сооружения

Расчет интегрированного ущерба в случае террористической атаки на тоннельные сооружения производится из предположения о максимальном ущербе сооружению — обвале тоннеля в результате его подрыва. Расчеты проводятся последовательно для всех составляющих $К^r_{инт}$ (см. формулу (1)). Важной особенностью ситуации, которая может возникнуть после террористической атаки на тоннель является то обстоятельство, что, как правило, в районах строительства тоннелей не существует путей объезда или они удалены на большие расстояния.

Подрыв тоннеля обычно не влечет за собой гибель большого количества транспортных средств. При обвале непосредственное повреждение получают только несколько автомобилей. Большая часть транспортных

средств пострадает от столкновений с идущими впереди машинами в зависимости от скорости и плотности грузопотока.

Число людей, отравившихся угарным газом, зависит от времени нахождения людей в тоннеле и наличия или отсутствия в нем принудительной вентиляции. Однако, прежде всего, людские потери будут зависеть от организационных мероприятий, предусмотренных на случай террористических актов (лимитирование параметров движения в тоннеле, оповещение водителей о порядке их действий в случае обвала тоннеля, наличие специальной сигнализации, эвакуационных камер и эвакуационных и средств других подобных им).

Перечисленные особенности позволяют предположить, что доминирующей составляющей потенциального ущерба при осуществлении террористической атаки на тоннель будет экономический ущерб от задержки грузов $К^{г}_{экон1}$, который главным образом определяется временем восстановления движения по тоннелю.

Ущерб $К_{л}$, определяемый численностью погибших и пострадавших в результате обвала тоннеля рассчитывается для двух отмеченных далее основных ситуаций.

Ситуация 1, когда реализованы перечисленные выше мероприятия по обеспечению безопасности. В этом случае в качестве стандартной нормы человеческих потерь при взрыве тоннеля целесообразно принять гибель двух человек, то есть ущерб $К^{об}_{л}$:

$$К_{л} = К^{об}_{л} = 50 \text{ млн. руб.} \qquad (12)$$

Ситуация 2 когда тоннель не снабжен системой обеспечения безопасности. В этом случае, если длина тоннеля не превышает 500 метров, то потери людей от отравления выхлопными газами не учитывается, а величина ущерба определяется принятым стандартом.

Если длина тоннеля более 500 метров, то в основу расчета ущерба в результате гибели людей положен следующий сценарий. После обвала тоннеля происходят столкновения транспорта и перекрытие части тоннеля столкнувшимися машинами. Проходит не менее 10 минут, пока люди в автомобилях сознают, что случилось, и начинают выбираться из тоннеля. Их движение затрудняют машины, перегородившие тоннель, в результате чего скорость выхода людей из тоннеля будет ограничена. Большинство столкнувшихся машин и машин затормозивших и находящихся в тоннеле не выключают двигатели и через 20–30 минут концентрация выхлопных газов становится критической. После этого люди не успевшие покинуть тоннель погибают от отравления выхлопными газами. Результат осуществления та-

кого сценария рассчитывается по формуле:

$$К_л = К_л^{об} + \left\{ \frac{L_{тп}(G_г S_г + G_п S_п)}{V} + \frac{L_т}{L_{тс}}(h_г S_г + h_п S_п)\left(1 - \frac{V_{вых}(T_{пр} - T_ш)}{L_т}\right)\right\} С_л, \quad (13)$$

где

- $К_л^{об}$ — ущерб от гибели людей в месте обвала тоннеля;
- $L_т$ — длина тоннеля в метрах;
- $L_{тп}$ — длина тормозного пути в метрах в зависимости от скорости транспортного средства;
- $L_{тс}$ — длина участка в метрах, занимаемая транспортным средством в колонне, остановившейся в тоннеле в результате обвала;
- $G_г$ — грузопоток — количество транспортных средств, проходящих какую либо точку трассы в минуту;
- $G_п$ — пассажиропоток — количество пассажирских транспортных средств, проходящих начальную линию мостового сооружения в минуту;
- $S_г$ — среднее количество людей в грузовых транспортных средствах;
- $S_п$ — среднее количество людей в пассажирских транспортных средствах;
- V — средняя скорость транспортных средств метров в мин.;
- $h_г$ — доля грузовых транспортных средств в колонне, образовавшейся в тоннеле после обвала;
- $h_п$ — доля пассажирских транспортных средств в колонне, образовавшейся в тоннеле после обвала;
- $V_{вых}$ — скорость движения людей выходящих из тоннеля метров в мин.;
- $T_{пр}$ — время, за которое концентрация выхлопных газов достигает смертельного уровня в мин;
- $T_ш$ — время шокового состояния людей в транспортных средствах до начала их выхода из тоннеля в мин.;
- $С_л$ — цена ущерба.

Ущерб $К_{экон}$ определяется как сумма стоимости выведенных из строя транспортных средств $К_{экон}^{тс}$, в результате обвала тоннеля и их столкновений после этого события, а также ущерба от уменьшения или прекращения грузопотока $К_{экон}^{г}$ в период от разрушения до восстановления тоннеля:

$$К_{экон} = К_{экон}^{тс} + К_{экон}^{г}. \quad (14)$$

$К^{тс}_{экон}$ рассчитывается по формуле (15) из условия того, что все транспортные средства, попавшие в аварию в тоннеле, выбывают из эксплуатации:

$$К^{тс}_{экон} = \frac{L_{тп} \bullet G}{V} \cdot С_{тс}, \quad (15)$$

где G — суммарный грузопоток — количество транспортных средств, проходящих начальную линию тоннеля в минуту.

Ущерб $К^{г}_{экон}$ рассчитывается по формуле:

$$К^{г}_{экон} = GT_{вос}С_{г}I_{шт}, \quad (16)$$

где

- $T_{вос}$ — время восстановления тоннеля;
- $I_{шт}$ — средняя величина штрафа в процентах от стоимости груза в зависимости от времени опоздания;
- $С_{г}$ — средняя стоимость груза в зависимости от структуры грузопотока.

Если имеется объезд тоннеля, то ущерб $К^{г}_{экон}$ рассчитывается так же, как и для мостовых сооружений по формулам (5) или (6).

Для оценки $К_{бал}$ необходимо использовать нормативы МЧС по восстановлению тоннелей.

Оценка экологического ущерба $К_{экол}$ исходит из того, что экологический ущерб при разрушении тоннеля может быть нанесен транспортными средствами с опасными грузами, которые могут быть разрушены непосредственно от обвала тоннеля или в результате столкновения с другими транспортными средствами после этого события. Для расчетов необходимо знать долю опасных грузов в грузопотоке q_i, где i — тип опасного груза. Кроме того, необходимо знать нормативы ущерба в денежном выражении от аварии транспортного средства с каждым типом опасного груза — $С^{i}_{эк}$. Такие нормативы имеются в МЧС. Тогда ущерб $К_{экол}$ может быть рассчитан по формуле:

$$К_{эк} = \sum_{i=1}^{I} \frac{L_{тп} \bullet G_{г}}{V} \cdot q_i С^{i}_{эк}, \quad (17)$$

где I — количество типов опасных грузов.

Необходимо отметить, что авария, повлекшая за собой высвобождение в тоннеле высокотоксичных веществ, может привести к трагическим последствиям. По этой причине для трасс с высокой долей опасных грузов в грузопотоке необходимо предусмотреть специальные мероприятия по транспортировке этих грузов через тоннели. В противном случае, всем

тоннелям, через которые транспортируется большое количество опасных грузов, необходимо присваивать высшую категорию опасности.

Суммарный ущерб, по которому производится категорирование автомобильных тоннелей, рассчитывается по формуле (2) с использованием таблицы (1).

Полученный результат сравнивается со значениями диапазонов ущербов в таблице 1 и объекту присваивается соответствующая категория.

Анализ опасности объектов инфраструктуры автомобильных дорог

Исследование инфраструктуры начинается с определения географических границ заданного региона. Далее в рамках указанных границ определяется множество ключевых объектов, определяющих, в значительной степени, пропускную способность дорог. Эта задача решается путем построения структурной модели предположительно опасных объектов. Такая модель должна отражать положение объекта в общей дорожной инфраструктуре. В случае, если объект имеет сложную структуру, то и отдельные его составляющие могут предположительно быть отнесены к числу опасных, то есть таких, от состояния которых зависит пропускная способность и безопасность дорожного объекта в целом.

Вопрос о необходимости представления тех или иных объектов в структурированном виде решается путем анализа возможностей и целесообразности выполнения таких подготовительных задач, как построение модели угроз и категорирование по отдельным его составляющим с последующим получением оценки по объекту в целом путем агрегирования полученных расчетов. На подготовительном этапе оценка опасности отдельных составляющих структурированных объектов проводится аналогично тому, как проводится анализ критичности по отдельным объектам. Для того, чтобы провести анализ опасности объектов необходимо построить модель угроз каждого объекта. Далее, следует построить модели событий рисков, при которых эти угрозы могут быть реализованы, для того, что бы оценить важность отдельных угроз и опасность объекта в целом.

Ведение в ПК «РискМенеджер-Анализ» каталога классов объектов позволяет выделить в отдельный набор предварительных процедур выполнение таких функций как создание БД всех известных угроз, мер защиты и требований по защите для каждого из классов объектов. Для решения этих задач разработан интерфейс, представленный на рис. 1.

Таким образом, по каждому классу объектов создается полное досье известных и потенциально применимых к нему угроз (модель угроз класса

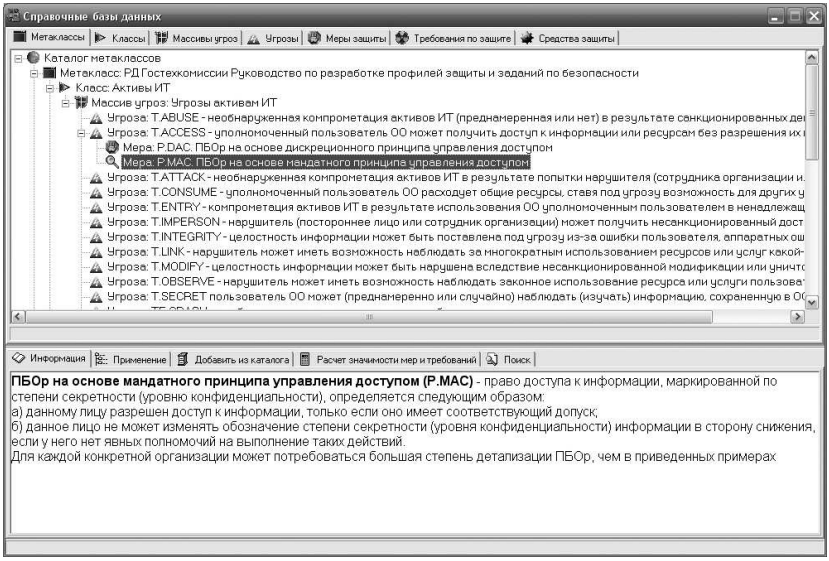


Рис. 1

объектов) и мер защиты от этих угроз (модель защиты класса объектов). Работа с интерфейсом, представленным на рис. 1, осуществляется с помощью контекстного меню. Актуализация БД этих досье, отражающая изменения, происходящие в действительности, является перманентной задачей системы обеспечения безопасности объекта.

Любой конкретный объект идентифицируется классом, к которому он принадлежит, а также его структурной координатой, то есть указанием его положения в структуре, в которую он входит.

Для наглядного отражения структурных координат всех объектов, опасность которых требуется оценить, в ПК «РискМенеджер-Анализ» предлагается построение *структурной модели* в рамках интерфейса, представленного на рис. 2.

Для задания структурных координат объектов и построения таким образом структурной модели могут использоваться четыре иерархически взаимосвязанных уровня: **М** (модель), **Р** (регион), **Л** (локальная среда), **П** (подсистема/процесс).

Для каждой структурной составляющей, в том числе для объектов, могут быть приведены и сохранены неограниченные по размеру и составу описания (закладка **Информация об объекте**). Построение и редактирование *структурной модели* (СМ) осуществляется с помощью контекстного

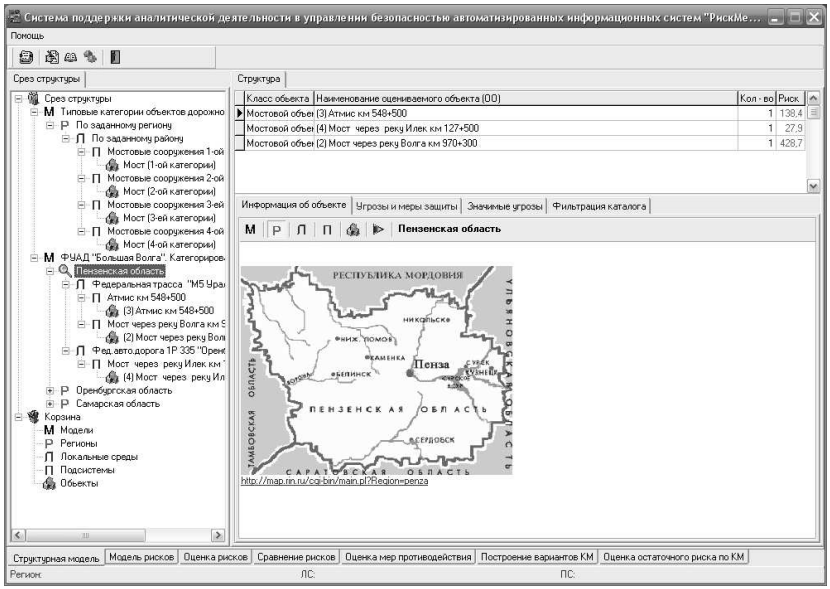


Рис. 2

меню (см. рис. 2). При определении нового объекта наряду с его названием требуется путем выбора из выпадающего списка указать его класс из числа определенных в БД Каталогов (см. рис. 3).

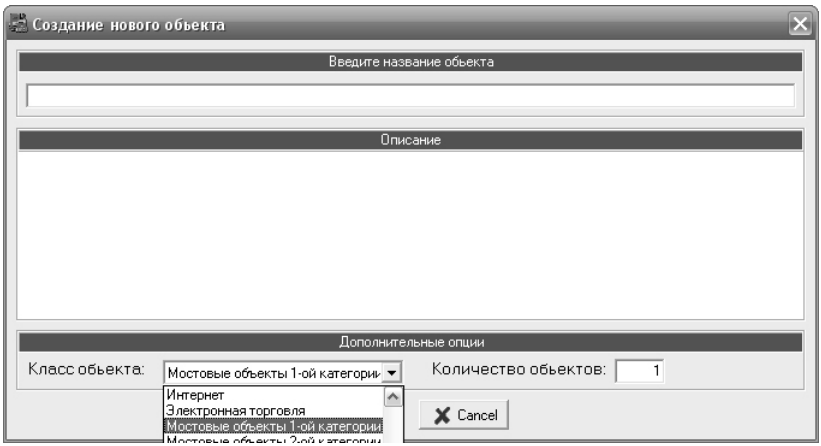


Рис. 3

Одновременно с построением СМ формируется *нормативная модель угроз* (НМУ). Название *нормативной* дано этой модели с тем, чтобы подчеркнуть, что в нее включены все известные угрозы, которые должны анализироваться на предмет их значимости с учетом конкретной среды и обстоятельств, в которых находится оцениваемый объект. Фрагмент НМУ по каждому из объектов СМ может быть просмотрен при переходе на закладку **Угрозы и меры защиты**. Полностью НМУ может быть получена в виде rtf-файла в редакторе Word (см. рис. 4) через опции печати контекстного меню.

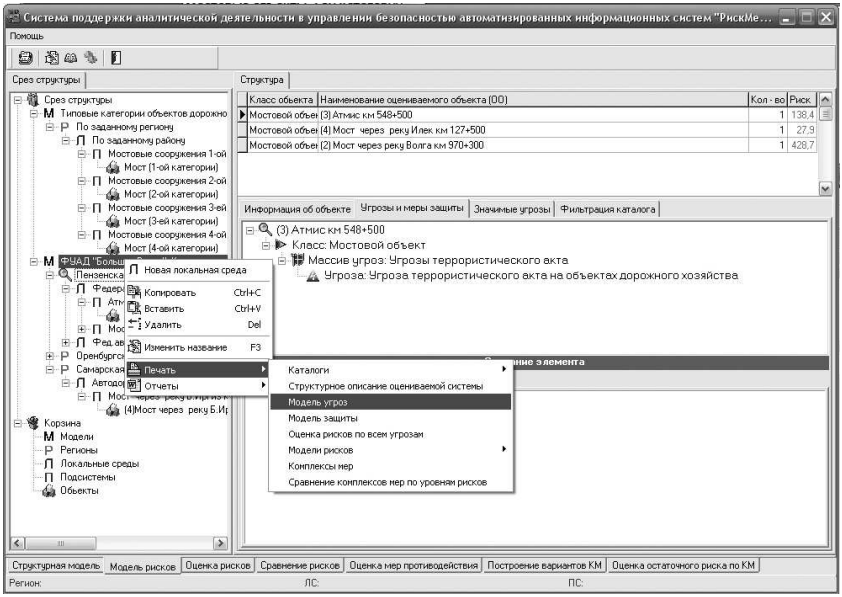


Рис. 4. Опции контекстного меню СМ

Следующим шагом должно стать обоснование актуальности (опасности) угроз через оценку возможного ущерба, который может быть связан с их возможной реализацией. Эту оценку опасности или возможного ущерба реализации угроз назовем *рискообразующим потенциалом* (РОП) угрозы.

Определять рискообразующий потенциал отдельных угроз предлагается путем построения моделей событий рисков (МСР), в результате которых могут быть реализованы эти угрозы. Для автоматизации решения

этой задачи предлагается интерфейс, представленный на рис. 5. Последовательность шагов построения МСР при этом следующая.

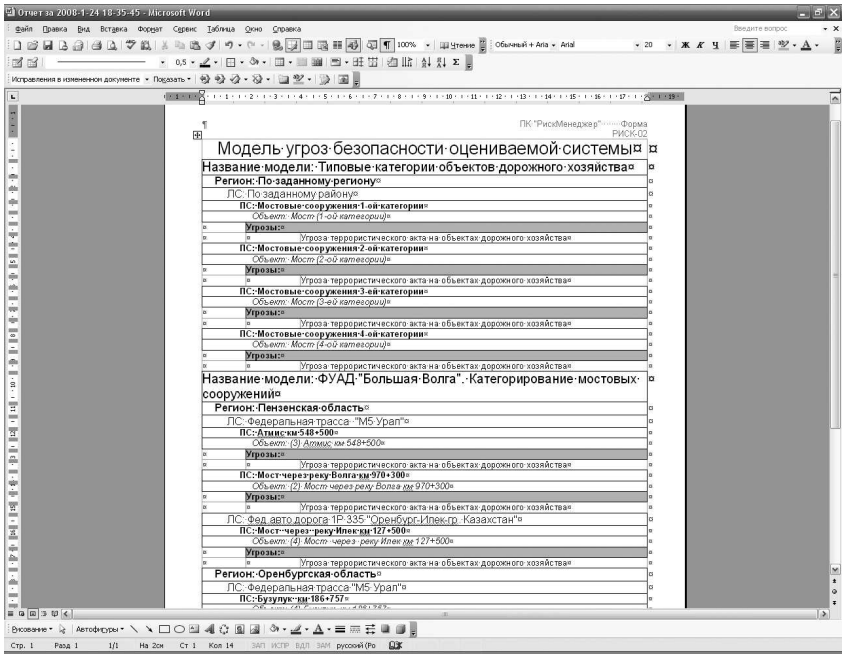


Рис. 5

Событию риска дается название. При необходимости более подробное описание представляется в поле комментария. Там же, в поле комментария могут быть представлены обоснования и доказательства оценок вероятности и возможного ущерба от этого события, либо даны гиперссылки на файлы в которых эти обоснования даны. В отдельных полях по каждому событию определяются оценки ущерба от события риска и вероятности этого события. Автоматически, как математическое ожидание на основе данных оценок вычисляется ожидаемый ущерб или рискообразующий потенциал, возникающий по причине возможности указанного события риска. Из массива (модели) всех идентифицированных угроз выбираются те, реализация которых может привести к событию риска. Они образуют множество угроз формирующих риск. На основе рассчитанной оценки рискообразующего потенциала события риска рассчитываются рискообразующие потенциалы каждой из угроз приводящих к событию риска. По возможности строятся все потенциально возможные модели событий рисков на

все угрозы, соответствующие моделям угроз объектов, чтобы доказать их значимость.

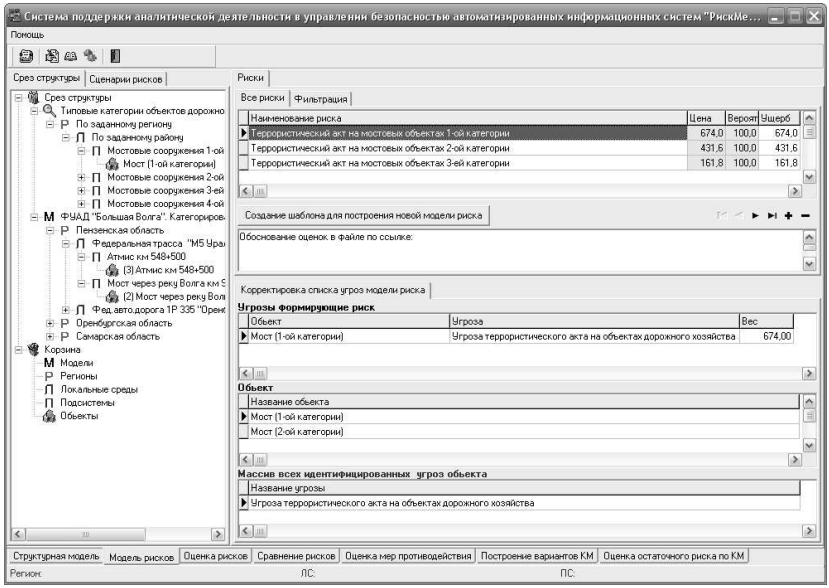


Рис. 6

Таким образом, только угрозы с ненулевым рискообразующим потенциалом попадают в число значимых, а их совокупность образует модели значимых угроз (МЗУ) по каждому из объектов и по организационным структурам, частью которых эти объекты являются.

На основании оценок рискообразующих потенциалов значимых угроз рассчитываются рискообразующие потенциалы объектов и структур, чьи МЗУ образуют значимые угрозы. Рискообразующие потенциалы объектов рассчитываются как суммы рискообразующих потенциалов угроз МЗУ этих объектов. Рискообразующие потенциалы структур уровня П (подсистемы и процессы), рассчитываются как суммы рискообразущих потенциалов объектов, входящих в (или образующие) соответствующие подсистемы или процессы.

Рискообразующие потенциалы структур уровня Л (локальных сред) рассчитывается как сумма рискообразующих потенциалов структур уровня П.

Рискообразующие потенциалы структур уровня Р (регионов) рассчитывается как сумма рискообразующих потенциалов структур уровня Л.

Рискообразующие потенциалы структур уровня M (структурных моделей) рассчитывается как сумма рискообразующих потенциалов структур уровня P.

Оценки рискообразующих потенциалов могут также трактоваться как оценки рисков по соответствующим структурным составляющим. Интерфейс представленный на рис. 7 дает возможность получить эти оценки в графическом виде. Возможно также получение отчетов в виде, представленном на рис. 8.

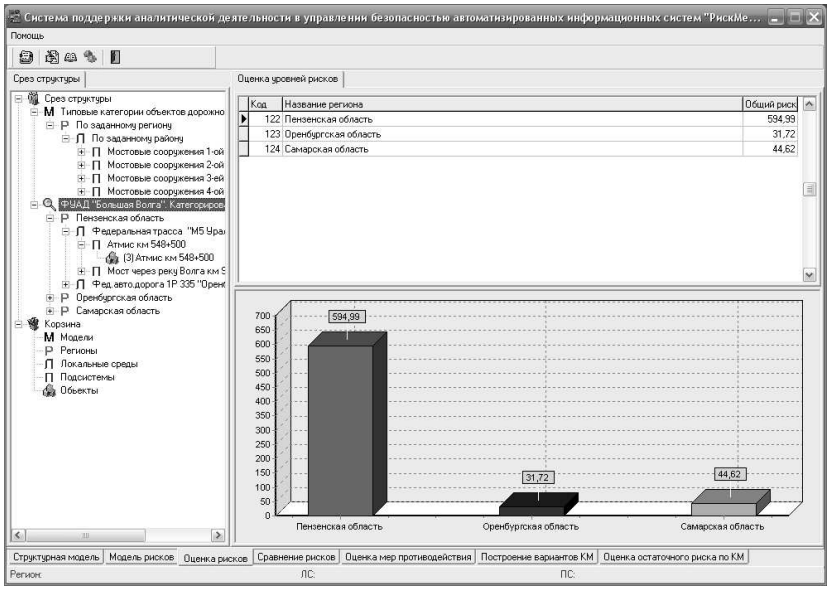


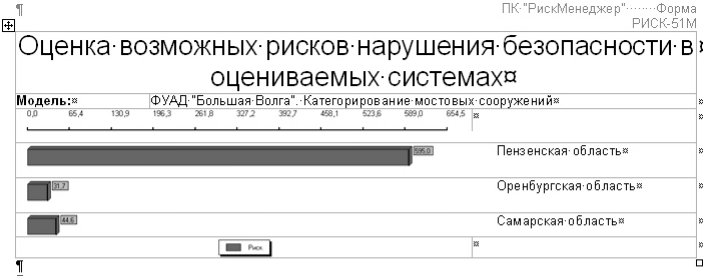
Рис. 7

Полученные таким образом оценки рисков или рискообразующих потенциалов по структурным составляющим дают возможность идентифицировать опасные составляющие.

Литература

[1] Концепция реализации основных положений Закона о транспортной безопасности. Минтранс РФ. 2008 г.

[2] Федеральный Закон «О транспортной безопасности» от 9.02.2007 № ФЗ-16.



	В целом
Название модели: ФУАД "Большая Волга". Категорирование мостовых сооружений	671,33
Регион: Пензенская область	594,99
ЛС: Федеральная трасса "М5 Урал"	567,10
ПС: Атмис км 548+500	138,37
Объект: (3) Атмис км 548+500	138,37
ПС: Мост через реку Волга км 970+300	428,73
Объект: (2) Мост через реку Волга км 970+300	428,73
ЛС: Фед.авто дорога 1Р-35 "Оренбург-Илек-гр. Казахстан"	27,89
ПС: Мост через реку Илек км 127+500	27,89
Объект: (4) Мост через реку Илек км 127+500	27,89
Регион: Оренбургская область	31,72
ЛС: Федеральная трасса "М5 Урал"	31,72
ПС: Бузулук км 186+757	31,72
Объект: (4) Бузулук км 186+757	31,72
Регион: Самарская область	44,62
ЛС: Автодорога М-32 "Самара-Б. Черниговка-гр. Респ. Казахстан"	44,62
ПС: Мост через реку Б.Иргиз км 102+400	44,62
Объект: (4) Мост через реку Б.Иргиз км 102+400 автомобильной дороги М-32 "Самара-Б.Черниговка-гр.Респ.Каз"	44,62

Рис. 8. Отчет с оценкой рисков

[3] *Кононов А. А.*, *Поликарпов А. К.* Автоматизация построения профилей защиты с использованием комплексной экспертной системы «АванГард» // Науч.-техн. информ. Сер. 1. 2003. № 8. С. 27–32.

[4] *Черешкин Д. С.*, *Кононов А. А.* Автоматизация построения профилей защиты // III Санкт-Петербургская Межрегиональная конференция «Информационная безопасность регионов России (ИБРР-2003)» (Санкт-Петербург, 25–27 ноября 2003 г.). Материалы конференции. Часть I. СПб.: 2003, с. 86.

[5] *Бурдин О. А.*, *Кононов А. А.* Метод оценки рискообразующих потенциалов в компьютеризированных организационных системах // НТИ. Сер. 1. 2004. № 2. С. 19–21.

МАТЕРИАЛЫ ЧЕТВЕРТОЙ МЕЖДУНАРОДНОЙ НАУЧНОЙ КОНФЕРЕНЦИИ ПО ПРОБЛЕМАМ БЕЗОПАСНОСТИ И ПРОТИВОДЕЙСТВИЯ ТЕРРОРИЗМУ

Том 2

Московский государственный университет им. М. В. Ломоносова,
30–31 октября 2008 г.

Подписано в печать 25.03.09 г. Формат $60 \times 90\ {}^{1}/_{16}$.
Бумага офсетная. Печать офсетная. Печ. л. 17,5.
Тираж 500 экз. Заказ №

Издательство Московского центра непрерывного математического образования
119002, Москва, Большой Власьевский пер., 11. Тел. (499) 241-74-83.

Отпечатано с готовых диапозитивов в ППП «Типография “Наука”»
121099, Москва, Шубинский пер., 6

Книги издательства МЦНМО можно приобрести в магазине «Математическая книга»,
Большой Власьевский пер., д. 11. Тел. (499) 241–72–85. E-mail: biblio@mccme.ru
http://www.mccme.ru/publications/