ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2008 **№ 2(2)**

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Адрес редакции: 634050, г. Томск, пр. Ленина, 36
E-mail: vestnik_pdm@mail.tsu.ru

В журнале публикуются результаты фундаментальных и прикладных научных исследований отечественных и зарубежных ученых, включая студентов и аспирантов, в области дискретной математики и ее приложений в криптографии, компьютерной безопасности, кибернетике, информатике, программировании и теории надежности.

Периодичность выхода журнала: 4 номера в год.

ООО «Издательство научно-технической литературы»
634050, Томск, пл. Ново-Соборная, 1, тел. (3822) 533-335

Редактор *Н.И. Шидловская*
Верстка *Д.В. Фортес*

Изд. лиц. ИД № 04000 от 12.02.2001. Подписано к печати 12.08.2008.
Формат 70 × 100 $^{1}/_{16}$. Бумага офсетная. Печать офсетная. Гарнитура «Таймс».
Усл. п. л. 16,04. Уч.-изд. л. 17,97. Тираж 300 экз. Заказ № 7.

Отпечатано в типографии «М-Принт», г. Томск, ул. Пролетарская, 38/1

СОДЕРЖАНИЕ[1]

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

МАТЕМАТИЧЕСКИЕ ОСНОВЫ НАДЕЖНОСТИ ВЫЧИСЛИТЕЛЬНЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

ПРИКЛАДНАЯ ТЕОРИЯ КОДИРОВАНИЯ

---

[1] Статьи этого номера представлены оргкомитетом VII Сибирской научной школы-семинара с международным участием «Компьютерная безопасность и криптография» – SIBECRYPT'08. Школа-семинар проведена совместно Томским госуниверситетом и Сибирским государственным аэрокосмическим университетом в сотрудничестве с Институтом криптографии, связи и информатики Академии ФСБ с 9 по 12 сентября 2008 года в г. Красноярске при финансовой поддержке РФФИ (грант № 08-07-06024-г).

CONTENTS

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

DOI 10.17223/20710410/2/1

УДК 621.391.1:004.7

РАССТОЯНИЕ ЕДИНСТВЕННОСТИ СЕМЕЙСТВА КООРДИНАТНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ, ПОЛУЧЕННЫХ УСЛОЖНЕНИЕМ ЛИНЕЙНЫХ РЕКУРРЕНТ НАД КОЛЬЦОМ ГАЛУА[1]

Д.Н. Былков

Московский государственный институт радиотехники, электроники и автоматики

E-mail: Bilkov@gmail.com

В настоящей работе обсуждается наличие эквивалентных последовательностей при усложнении координатных ЛРП над кольцом Галуа, а также рассмотрен вопрос о минимальной длине, на которой выходные последовательности будут различимы.

***Ключевые слова*:** *координатные ЛРП, кольцо Галуа, эквивалентные состояния, расстояние единственности.*

Пусть $R = GR(q^n, p^n)$ – кольцо Галуа характеристики p^n, состоящее из q^n элементов [1], $n > 1$, $q = p^r$, e – единица. Для кольца Галуа R поле $\overline{R} = R/pR$ будем называть полем вычетов. Естественный эпиморфизм $R \to \overline{R}$ индуцирует эпиморфизм колец многочленов $R[x] \to \overline{R}[x]$. Образ многочлена $A(x) = \sum a_i x^i \in R[x]$ будем обозначать $\bar{A}(x)$: $\bar{A}(x) = \sum \bar{a}_i x^i \in \overline{R}[x]$.

Подмножество $B = \{b_0, \dots, b_{q-1}\}$ называется координатным множеством кольца R, если его элементы образуют полную систему вычетов по модулю идеала pR. В [1] показано, что каждый элемент $a \in R$ однозначно представляется в виде $a = a_0 + pa_1 + \dots + p^{n-1}a_{n-1}$, $a_s \in B$, $0 \le s \le n-1$, называемом разложением элемента a в координатном множестве B.

Пусть $l, t \in \mathbb{N}$ и t удовлетворяет условиям $1 \le t \le l$. Произвольную функцию $h: R^l \to B$ от переменных $x_1, \dots, x_l$ можно рассматривать как функцию от переменных x_{ij}, $0 \le i \le n-1$, $1 \le j \le l$, отображающую B^{nl} в B, где величины $x_{0j}, \dots, x_{n-1\,j}$ суть коэффициенты из разложения переменной x_j в координатном множестве B. Скажем, что отображение h биективно по переменной $x_{n-1\,t}$, если при произвольной фиксации $nl-1$ переменных

$$(x_{0\,1}, \dots, x_{n-1\,1}, \dots, x_{0\,t}, \dots, x_{n-2\,t}, x_{0\,t+1}, \dots, x_{n-1\,t+1}, \dots, x_{0\,l}, \dots, x_{n-1\,l}) \in B^{nl-1}$$

функция $\hat{h} = h(x_{0\,1}, \dots, x_{n-2\,t}, x, x_{0\,t+1}, \dots, x_{n-1\,l})$ – биекция.

Всюду далее $F(x)$ – унитарный многочлен над R, через $L_R(F)$ будем обозначать множество всех линейных рекуррентных последовательностей над R с характеристическим многочленом $F(x)$. Каждой последовательности u из $L_R(F)$ сопоставим последовательности $u_0, \dots, u_{n-1}$, получающиеся из разложения знаков последовательности u в координатном множестве B: $u(i) = u_0(i) + pu_1(i) + \dots + p^{n-1}u_{n-1}(i)$, $u_s(i) \in B$, $0 \le s \le n-1$, $i \ge 0$.

Зафиксируем параметры $l \in \mathbb{N}$, $1 \le t \le l$, и $s_1, \dots, s_l \in \mathbb{N}$. Рассмотрим алгоритм A, сопоставляющий каждой из последовательностей $u \in L_R(F)$ выходную последовательность $\tilde{u} = A(u)$ по правилу: $\tilde{u} = h(u(i+s_1), \dots, u(i+s_l))$, $i \ge 0$, где отображение h биективно по переменной $x_{n-1\,t}$.

Будем говорить, что алгоритм A не допускает эквивалентных состояний, если существует натуральное L со свойством

$$\forall\ u, v \in L_R(F) \quad (u \ne v) \Rightarrow ((\tilde{u}(0), \dots, \tilde{u}(L-1)) \ne (\tilde{v}(0), \dots, \tilde{v}(L-1))). \qquad (1)$$

В этом случае минимум $\mathrm{Ud}(F, A)$ значений L со свойством (1) назовем расстоянием единственности алгоритма A. В противном случае будем считать $\mathrm{Ud}(F, A) = \infty$.

Определим норму элемента $a \in R$ и норму последовательности $y \in R^{<1>}$ следующим образом:

$$\|a\| = \max\{0 \le t \le n \mid a \in p^t R\},\ \|y\| = \max\{0 \le t \le n \mid y \in p^t R^{<1>}\}.$$

[1] Работа выполнена при финансовой поддержке Совета поддержки научных школ при Президенте РФ (номер проекта НШ – 8564.2006.10).

Положим $L_R(F)^* = \{u \in L_R(F) : \|u\| = 0\}$. Будем говорить, что многочлен $F(x) \in R[x]$ реверсивный, если $F(0) \in R^*$. Назовем многочлен $F(x) \in R[x]$ многочленом Галуа, если его образ $\overline{F}(x) \in \overline{R}[x]$ неприводим над $\overline{R}$. При условии $T(F) = p^{n-1}(q^m - 1)$, $m = \deg F(x)$, скажем, что $F(x)$ – многочлен максимального периода (ММП) над R.

Теорема 1. Пусть существует натуральное L, такое, что для любой последовательности $u \in L_R(F)^*$ для каждого $0 \le k \le n-1$ найдется $0 \le i_0 \le L-1$ со свойством

$$u(i_0 + s_j) = 0, \text{ для всех } j \in \{1, \ldots, l\} \setminus \{t\},$$
$$\|u(i_0 + s_t)\| = k.$$

Тогда верно неравенство $\mathrm{Ud}(F, A) \le L$.

Условия теоремы 1 выполняются, например, если $l = 1$ и $F(x)$ – многочлен максимального периода. В [1] показано, что на цикле линейной рекурренты максимального периода появятся все элементы кольца R, то есть справедливо неравенство $L \le T(F)$. В случае кольца $\mathbb{Z}_4$ можно указать более точные оценки величины L.

Теорема 2. Пусть $F(x) = x^m - x - 1 \in \mathbb{Z}_4[x]$, где $m > 2$. Тогда для любой последовательности $u \in L_{\mathbb{Z}_4}(F)^*$, для всех $z \in \{1, 3\}$ существует $i_0 \le 4m - 2$, такое, что $u(i_0) = z$. Причем указанная оценка достижима.

Теорема 3. Пусть $F(x) = x^m - x^k - 1$ – многочлен Галуа над кольцом $\mathbb{Z}_4$, где $1 < k < m$. Тогда для любой последовательности $u \in L_{\mathbb{Z}_4}(F)^*$, для всех $z \in \{1, 3\}$ существует $i_0 \le 3mk + 2m - 2k - 1$, такое, что $u(i_0) = z$.

Следующие результаты дают достаточные условия конечности величины $\mathrm{Ud}(F, A)$.

Теорема 4. Пусть $F(x)$ – многочлен Галуа степени m над кольцом R, удовлетворяющий условию

$$T(F) \ge p^{2(n-1)}(q^{nl} - 1)q^{m/2}.$$

Пусть также α – корень многочлена $F(x)$ в расширении $S = GR(q^{mn}, p^n)$ кольца R. Тогда при условии, что система $\alpha^{s_1}, \ldots, \alpha^{s_l}$ линейно независима над R, верно неравенство $\mathrm{Ud}(F, A) < \infty$.

Нетрудно заметить, что результат теоремы 4 нетривиален при выполнении неравенства $m \ge 2nl(1 + o(1))$.

Теорема 5. Пусть $l = t = 1$ и $F(x) = F_1(x)\,F_2(x)$, где $F_1(x)$, $F_2(x)$ – унитарные и реверсивные многочлены над R, такие, что $(T(F_1), T(F_2)) = 1$. Пусть также для произвольной последовательности $u \in L_R(F_s)^*$, $s = 1, 2$, для каждого $0 \le k \le n$ найдется $i_0 < T(F)$ со свойством $\|u(i_0)\| = k$. Тогда верно неравенство $\mathrm{Ud}(F, A) < \infty$.

Теорема 6. Пусть $l = t = 1$ и $F(x) = F_1(x)\,F_2(x)$, где $F_1(x)$, $F_2(x)$ – взаимно простые, унитарные, реверсивные многочлены Галуа над R, степеней m и k соответственно, такие, что

$$d_0 \ge p^{2(n-1)}q^{m/2},\ d_1 \ge p^{2(n-1)}q^{k/2},$$

где $t_i = T(F_i)$, $d_i = t_i / (t_1, t_2)$, $i = 1, 2$. Пусть также выполнено равенство $(T(\overline{F}_1), T(\overline{F}_2)) = 1$. Тогда верно неравенство $\mathrm{Ud}(F, A) < \infty$.

В некоторых случаях удается получить более точные оценки величины $\mathrm{Ud}(F, A)$. Всюду далее $R = \mathbb{Z}_4$, $l = t = 1$, $h(x) = x_{11}$ и вместо $\mathrm{Ud}(F, A)$ будем писать $\mathrm{Ud}(F)$. В [2] показано, что если $F(x)$ – многочлен максимального периода степени m, то для любой u из $L_R(F)^*$ выполнено равенство $\mathrm{rank}(u_1) = m(m+3)/2$. И поэтому $\mathrm{Ud}(F) \le m(m+3)/2$. Ниже приводятся оценки расстояния единственности для трехчленов вида $F(x) = x^m + ax^k + b$, $(a, b) \ne (1, 1)$.

Вид $F(x)$	Дополнительные условия	Теоретически полученная верхняя оценка $\mathrm{Ud}(F)$	Точные значения $\mathrm{Ud}(F)$ для $m \le 18$
$x^m - x^k - 1$	$\overline{F}(x)$ примитивный, $k < (m^2 - m)/(2m - 3)$	$m(3 + k) - 2k + 1$	$\mathrm{Ud}(F) \le 3m$
	$\overline{F}(x)$ примитивный, $k \ge (m^2 - m)/(2m - 3)$	$m(3 + m - k) - m + k + 1$	
	k делит m	$4m - 2k + 1$	
	$m - k$ делит m	$3m + k + 1$	
$x^m - 3x^k - 1$	$\overline{F}(x)$ примитивный, $k < (m^2 - 2m)/(2m - 3)$	$m(3 + k) - k + 1$	$\mathrm{Ud}(F) \le 3m$
	$\overline{F}(x)$ примитивный, $k \ge (m^2 - 2m)/(2m - 3)$	$m(3 + m - k) - 2m + 2k + 1$	
	k делит m	$4m - k + 1$	
	$m - k$ делит m	$2m + 2k + 1$	
$x^m - x^k - 3$	$\overline{F}(x)$ примитивный, $k < (m - 1)/2$	$m(3 + k) - k + 1$	$\mathrm{Ud}(F) \le 4m$
	$\overline{F}(x)$ примитивный, $k \ge (m - 1)/2$	$m(3 + m - k) - m + k + 1$	
	k делит m	$4m - k + 1$	
	$m - k$ делит m	$3m + k + 1$	

Оказывается, для большого числа трехчленов максимального периода теоретическая оценка расстояния единственности существенно меньше $m(m + 3) / 2$. Имеющиеся результаты точного вычисления на ПК расстояния единственности для указанных многочленов позволяют выдвинуть гипотезу о том, что для таких многочленов $\mathrm{Ud}(F) \le 4m$.

Показано, что если $F(x) \in \{x^{2+2i} + x^i + 1, x^{2+3i} + x + 1, x^{4+3i} + x^2 + 1\}$, $i \ge 0$, то $\mathrm{Ud}(F) = \infty$.

В отдельных случаях можно указать точное значение величины $\mathrm{Ud}(F)$.

Теорема 7. Пусть $R = \mathbb{Z}_4$, $l = t = 1$, $h(x) = x_{11}$ и $F(x) = x^m - x - 1$. Тогда справедливо равенство $\mathrm{Ud}(F) = 3m$.

Автор выражает глубокую благодарность А.А. Нечаеву за помощь в проведении исследования и ценные советы по оформлению данной работы.

ЛИТЕРАТУРА

1. *Кузьмин А.С.*, *Куракин В.Л.*, *Нечаев А.А.* Кольца Галуа в приложениях к кодам и рекуррентам // Труды Академии криптографии РФ. Тема 26. М., 1998.
2. *Нечаев А.А.* Код Кердока в циклической форме // Дискретная математика. 1989. Т. 4. № 1. С. 123 – 139.

2008 Теоретические основы прикладной дискретной математики № 2(2)

DOI 10.17223/20710410/2/2

УДК 519.682

О РЕШЕНИИ СИСТЕМ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ, АССОЦИИРОВАННЫХ С КОНТЕКСТНО-СВОБОДНЫМИ ЯЗЫКАМИ

О.И. Егорушкин, Д.А. Калугин-Балашов, К.В. Сафонов

Красноярский государственный аграрный университет,
Сибирский федеральный университет, *г. Красноярск*

E-mail: safonovkv@rambler.ru

Рассмотрены системы алгебраических (полиномиальных) уравнений над кольцом, некоммутативным относительно умножения. Получено условие разрешимости таких систем в виде формальных степенных рядов. Рассмотрены системы линейных алгебраических уравнений, для которых исследована возможность понижения порядка систем. Данные системы обобщают свойства систем уравнений, определяющих контекстно-свободные и линейные языки.

Ключевые слова: *контекстно-свободные языки*, *системы алгебраических уравнений*, *некоммутативное кольцо*, *коммутативный образ*, *граф инцидентности.*

Обычно формальным языком L называют множество цепочек в алфавите $\{x_1,\ldots, x_n\}$, выделенных с помощью конечного набора правил. Выделенные цепочки, принадлежащие свободной полугруппе $\{x_1,\ldots, x_n\}^*$, называются при этом словами (над алфавитом) либо правильно построенными предложениями (над словарем), либо грамматически правильными предложениями. Конечное множество правил, с помощью которых выделяются цепочки, называют грамматикой. Таким образом, формальный язык определяется совокупностью соответствующих правил и способом выделения цепочек с помощью этих правил.

Практически важный класс формальных языков образуют контекстно-свободные языки (кс-языки), поскольку они являются адекватным средством моделирования естественных языков, а также языков программирования [1 – 3].

Рассмотрим играющее роль словаря конечное множество $X = \{x_1, \ldots, x_n\}$, состоящее из слов x_i языка и называемое терминальным множеством, а также $Z = \{z_1, \ldots, z_m\}$ – множество вспомогательных символов z_j, необходимых для задания грамматических правил, называемое нетерминальным множеством. Обозначим $W^* = (X \cup Z)^* = (\{x_1, \ldots, x_n\} \cup \{z_1, \ldots, z_n\})^*$ – свободную полугруппу относительно операции конкатенации; ее элементами являются произвольные цепочки, составленные из элементов «расширенного» алфавита $\{x_1, \ldots, x_n, z_1, \ldots, z_n\}$ – соответствующая свободная полугруппа относительно операции.

Дополним её операцией формального сложения «+» мономов из множества W^* (вместо суммы можно взять объединение «$\cup$», следуя [1]), а также коммутативной операцией умножения мономов на (целые) числа. Таким образом, можно рассматривать не только многочлены, но и формальные степенные ряды с числовыми (как правило, целыми) коэффициентами от некоммутативных переменных.

Кс-грамматика есть совокупность правил подстановки, которые каждому нетерминальному символу z_i ставят в соответствие некоторый моном от терминальных и нетерминальных символов:

$$z_j \to f_{j1}(x,z),\ldots, z_j \to f_{jq\,j}(x,z)\,,$$

при этом z_1– особый, выделенный символ – начальный символ предложения (или программы). Правилам подстановки ставится в соответствие [3] система полиномиальных уравнений: каждому вспомогательному символу z_j, содержащемуся в левой части подстановки, сопоставляется уравнение $z_j = p_j(x, z)$, где

$$p_j(x, z) = f_{j1}(x, z) + \ldots + f_j q_j(x, z).$$

Таким образом, кс-грамматике соответствует система полиномиальных уравнений

$$z_j = p_j(x, z), j = 1, \ldots, m, \tag{1}$$

и кс-языком называется [3, 4] первая компонента z_j решения $(z_1(x), \ldots, z_m(x))$ этой системы полиномиальных уравнений, получаемого методом последовательных приближений:

$$z_i^{(k+1)} = p_i(x, z^{(k)}),\ z_i^{(0)} = 0,\ i = 1, \ldots, m,$$

где $z^{(k)} = (z_i^{(k)},\ldots, z_m^{(k)})$, 0 – нулевой моном, такой, что $0\cdot u = u\cdot 0 = 0$ для любого монома u.

В результате итераций каждая компонента z_j выражается формальным степенным рядом, причем кс-язык и есть тот формальный степенной ряд, который представляет выделенный символ z_1:

$$z_1 = \sum_i < z_1, w_i > w_i\ . \tag{2}$$

Здесь $< z_1, w_j >$ – числовой коэффициент, с которым моном w_j от некоммутативных переменных входит в ряд z_1. Мономы w_j являются грамматически правильными предложениями, которые могут быть построены в данном языке из слов $x_1, \ldots, x_n$ этого языка, а весь ряд (2), т.е. формальная сумма всех правильных предложений, и является данным кс-языком. Построение системы уравнений (1) по грамматическим правилам языка приведено выше, причем условие того, что язык является контекстно-свободным, состоит также и в том, что многочлены $p_j(x, z)$ не содержат мономов z_j и e, где e – пустая цепочка. Таким образом, существование и единственность решения системы (1) обеспечивается её специфической структурой в совокупности с методом последовательных приближений.

Рассмотрим общий случай, а именно ассоциированную с контекстно-свободными языками произвольную систему алгебраических (полиномиальных) уравнений:

$$q_i(x, z) = 0,\ i=1, \ldots, m. \tag{3}$$

Решением этой системы назовем выражение символов z_i в виде формальных степенных рядов от x, подстановка которых в многочлены $q_i(x, z)$ обращает их в нуль. Нас интересует, каковы условия, при которых система (3) имеет такое решение, и способ получения искомых рядов.

Поставим в соответствие формальному степенному ряду (многочлену) ряд (многочлен) с комплексными переменными, задав отображение терминальных x_i и нетерминальных z_i символов из множества $X \cup Z$ в множество комплексных переменных, причем оставляем за ними прежние обозначения, тогда $(x, z) \in C_{x,z}^{m+n}$. Таким образом, получаем фиксированный гомоморфизм, который ставит в соответствие формальному ряду (2) его *коммутативный образ* – степенной ряд от комплексных переменных

$$ci(r) = \sum_k a_k z^k,$$

где

$$a_k z^k = a_{k_1,\ldots,k_n} z_1^{k_1} \ldots z_1^{k_n},$$

$$a_k = \sum_{\#x_1(w_i)=k_1,\ldots,\#x_n(w_i)=k_n} < r, w_i >,$$

символ $\#c(d)$ означает число вхождений символа c в моном d. Коммутативный образ кс-языка является алгебраической функцией, голоморфной в некоторой окрестности нуля, как показывают оценки коэффициентов степенного ряда.

Имеет место следующая

Теорема 1. Если выполнено условие

$$J = \det\left(\frac{\partial(ci(q_i(0,0))}{\partial z_j}\right) \neq 0, \tag{4}$$

то система (3) имеет единственное решение в виде формальных степенных рядов, выражающих вектор-функцию z через компоненты вектора x.

Для доказательства заметим, что если формальные ряды $z = z(x)$ удовлетворяют системе уравнений $q(x, z) = 0$, то выполняется условие

$$ci(q(x, ci(z(x)))) = 0.$$

И хотя обратное, вообще говоря, не верно, условие (4) для якобиана J обеспечивает существование и единственность решения системы $ci(q(x, z)) = 0$; обозначим это решение $ci(z(x))$. Учитывая, что условие (4) означает линейную независимость градиентов коммутативных образов многочленов, а также метод мономиальных меток, предложенный в [4], получаем отсюда, что это решение действительно является коммутативным образом некоторого решения исходной некоммутативной системы.

Далее, в силу невырожденности матрицы Якоби в начале координат, можно сделать линейную замену переменных z_i, в результате которой эта матрица становится единичной, что и приводит систему (3) к виду (1), в результате чего её можно решать методом последовательных приближений. Искомые ряды равны линейной комбинации получаемых рядов.

Рассмотрим теперь системы линейных алгебраических уравнений над некоммутативным кольцом, тесно связанные с линейными языками [1]. Возможные методы решения этих систем в виде рядов должны учитывать как некоммутативность переменных по умножению, так и отсутствие операции деления в кольце W. Нам понадобятся некоторые определения.

L-операторами называются отображения $L : W \to W$, действие которых заключается в умножении элемента кольца W слева и справа на некоторые мономы l и r: $L(z) = l \cdot z \cdot r. L(z) = l \cdot z \cdot r.$

Основные свойства L -операторов:

1) $L(\alpha z_1 + \beta z_2) = \alpha L(z_1) + \beta L(z_2)$ (*линейность*);

2) если $L_1(z) = l_1 z r_1$ и $L_2(z) = l_2 z r_2$, то $L_1(L_2(z)) = L_1(l_2 z r_2) = l_1 l_2 z r_2 r_1 = L_3(z)$ (*композиция L-операторов является L-оператором*).

Уравнение $f(z_1, \ldots, z_m) = g(z_1, \ldots, z_m)$ называется *приведенным*, если не существует такого L-оператора R, что $f(z_1, \ldots, z_m) = R(f_1(z_1, \ldots, z_m))$ и $g(z_1, \ldots, z_m) = R(g_1(z_1, \ldots, z_m))$. Если уравнение не является приведенным, то его можно привести к эквивалентному, отбросив оператор R.

Система линейных алгебраических уравнений порядка n над некоммутативным кольцом имеет вид

$$\begin{cases} \sum\limits_k l_{1,1,k} L_{1,1,k}(z_1) + \sum\limits_k l_{1,2,k} L_{1,2,k}(z_2) + \ldots + \sum\limits_k l_{1,n,k} L_{1,n,k}(z_n) = \sum\limits_k \varphi_{1,k} \Phi_{1,k}, \\ \sum\limits_k l_{2,1,k} L_{2,1,k}(z_1) + \sum\limits_k l_{2,2,k} L_{2,2,k}(z_2) + \ldots + \sum\limits_k l_{2,n,k} L_{2,n,k}(z_n) = \sum\limits_k \varphi_{2,k} \Phi_{2,k}, \\ \ldots \\ \sum\limits_k l_{n,1,k} L_{n,1,k}(z_1) + \sum\limits_k l_{n,2,k} L_{n,2,k}(z_2) + \ldots + \sum\limits_k l_{n,n,k} L_{n,n,k}(z_n) = \sum\limits_k \varphi_{n,k} \Phi_{n,k}, \end{cases}$$

где $l_{i,j,k}$, $\varphi_{i,k}$ – действительные коэффициенты; $L_{i,j,k}(z)$ – L-операторы; $\Phi_{j,k}$ – мономы. Можно считать, что все уравнения системы являются приведенными.

Покажем, как исключать из данной системы одно неизвестное и одно уравнение. Исключаем, например, неизвестное z_n. Для каждого уравнения системы оставим все слагаемые с z_n в левой части, а все остальные слагаемые перенесем в правую часть, переписав систему в виде

$$\begin{cases} \sum\limits_k l_{1,n,k} L_{1,n,k}(z_n) = \sum\limits_k \varphi_{1,k} \Phi_{1,k} - \sum\limits_k l_{1,1,k} L_{1,1,k}(z_1) - \ldots - \sum\limits_k l_{1,n-1,k} L_{1,n-1,k}(z_{n-1}), \\ \sum\limits_k l_{2,n,k} L_{2,n,k}(z_n) = \sum\limits_k \varphi_{2,k} \Phi_{2,k} - \sum\limits_k l_{2,1,k} L_{2,1,k}(z_1) - \ldots - \sum\limits_k l_{2,n-1,k} L_{2,n-1,k}(z_{n-1}), \\ \ldots \\ \sum\limits_k l_{n,n,k} L_{n,n,k}(z_n) = \sum\limits_k \varphi_{n,k} \Phi_{n,k} - \sum\limits_k l_{n,1,k} L_{n,1,k}(z_1) - \ldots - \sum\limits_k l_{n,n-1,k} L_{n,n-1,k}(z_{n-1}). \end{cases}$$

Построим граф с n вершинами, i-я вершина которого соответствует $\sum\limits_k l_{i,n,k} L_{i,n,k}(z_n)$. Вершины i и j инцидентны, если существуют такие L-операторы R_{ij} и R_{ji}, что выполняется равенство

$$R_{ij}\left(\sum_k l_{i,n,k} L_{i,n,k}(z_n)\right) = R_{ji}\left(\sum_k l_{j,n,k} L_{j,n,k}(z_n)\right).$$

При этих обозначениях и условиях имеет место

Теорема 2. Порядок n системы линейных алгебраических уравнений можно понизить до $n - 1$, если существует вершина r, смежная всем остальным вершинам. Каждому ребру, инцидентному вершине r, соответствует одно уравнение новой системы. Очевидно, что таких ребер будет $n - 1$, следовательно, и порядок новой системы будет $n - 1$ (при выполнении условий теоремы для одной вершины граф инцидентности имеет вид, например, такой, как на рис. 1).

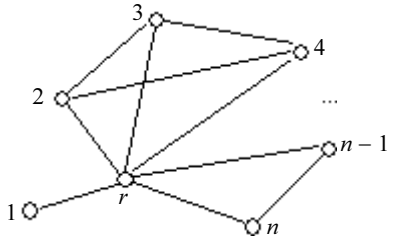

Рис. 1. Граф инцидентности вершины r

В случае выполнения условий теоремы 2 полученная система запишется в виде

$$\begin{cases} R_{r,1}\left(\sum\limits_k \varphi_{r,k} \Phi_{r,k} - \sum\limits_k l_{r,1,k} L_{r,1,k}(z_1) - \ldots - \sum\limits_k l_{r,n-1,k} L_{r,n-1,k}(z_{n-1})\right) = \\ = R_{1,r}\left(\sum\limits_k \varphi_{1,k} \Phi_{1,k} - \sum\limits_k l_{1,1,k} L_{1,1,k}(z_1) - \ldots - \sum\limits_k l_{1,n-1,k} L_{1,n-1,k}(z_{n-1})\right); \\ R_{r,2}\left(\sum\limits_k \varphi_{r,k} \Phi_{r,k} - \sum\limits_k l_{r,1,k} L_{r,1,k}(z_1) - \ldots - \sum\limits_k l_{r,n-1,k} L_{r,n-1,k}(z_{n-1})\right) = \\ = R_{2,r}\left(\sum\limits_k \varphi_{2,k} \Phi_{2,k} - \sum\limits_k l_{2,1,k} L_{2,1,k}(z_1) - \ldots - \sum\limits_k l_{2,n-1,k} L_{2,n-1,k}(z_{n-1})\right); \\ \ldots \\ R_{r,n}\left(\sum\limits_k \varphi_{r,k} \Phi_{r,k} - \sum\limits_k l_{r,1,k} L_{r,1,k}(z_1) - \ldots - \sum\limits_k l_{r,n-1,k} L_{r,n-1,k}(z_{n-1})\right) = \\ = R_{n,r}\left(\sum\limits_k \varphi_{n,k} \Phi_{n,k} - \sum\limits_k l_{n,1,k} L_{n,1,k}(z_1) - \ldots - \sum\limits_k l_{n,n-1,k} L_{n,n-1,k}(z_{n-1})\right). \end{cases}$$

Используя свойство линейности L-оператора, преобразуем эту систему к виду

$$\begin{cases}
\sum\limits_k \varphi_{r,k} R_{r,1}\left(\Phi_{r,k}\right) - \sum\limits_k l_{r,1,k} R_{r,1}\left(L_{r,1,k}\left(z_1\right)\right) - \ldots - \sum\limits_k l_{r,n-1,k} R_{r,1}\left(L_{r,n-1,k}\left(z_{n-1}\right)\right) = \\
= \sum\limits_k \varphi_{1,k} R_{1,r}\left(\Phi_{1,k}\right) - \sum\limits_k l_{1,1,k} R_{1,r}\left(L_{1,1,k}\left(z_1\right)\right) - \ldots - \sum\limits_k l_{1,n-1,k} R_{1,r}\left(L_{1,n-1,k}\left(z_{n-1}\right)\right); \\
\sum\limits_k \varphi_{r,k} R_{r,2}\left(\Phi_{r,k}\right) - \sum\limits_k l_{r,1,k} R_{r,2}\left(L_{r,1,k}\left(z_1\right)\right) - \ldots - \sum\limits_k l_{r,n-1,k} R_{r,2}\left(L_{r,n-1,k}\left(z_{n-1}\right)\right) = \\
= \sum\limits_k \varphi_{2,k} R_{2,r}\left(\Phi_{2,k}\right) - \sum\limits_k l_{2,1,k} R_{2,r}\left(L_{2,1,k}\left(z_1\right)\right) - \ldots - \sum\limits_k l_{2,n-1,k} R_{2,r}\left(L_{2,n-1,k}\left(z_{n-1}\right)\right); \\
\ldots \\
\sum\limits_k \varphi_{r,k} R_{r,n}\left(\Phi_{r,k}\right) - \sum\limits_k l_{r,1,k} R_{r,n}\left(L_{r,1,k}\left(z_1\right)\right) - \ldots - \sum\limits_k l_{r,n-1,k} R_{r,n}\left(L_{r,n-1,k}\left(z_{n-1}\right)\right) = \\
= \sum\limits_k \varphi_{n,k} R_{n,r}\left(\Phi_{n,k}\right) - \sum\limits_k l_{n,1,k} R_{n,r}\left(L_{n,1,k}\left(z_1\right)\right) - \ldots - \sum\limits_k l_{n,n-1,k} R_{n,r}\left(L_{n,n-1,k}\left(z_{n-1}\right)\right).
\end{cases}$$

Так как композиция L-операторов есть L-оператор, то полученная система является системой линейных алгебраических уравнений (над некоммутативным кольцом), её порядок равен $n - 1$. Тем самым порядок системы понижен на единицу, исключена неизвестная z_n.

ЛИТЕРАТУРА

1. *Глушков В.М.*, *Цейтлин Г.Е.*, *Ющенко Е.Л.* Алгебра, языки, программирование. Киев: Наукова думка, 1974.
2. *Семёнов А.Л.* Алгоритмические проблемы для степенных рядов и контекстно-свободных грамматик // Докл. АН СССР. 1973. Т. 212. С. 50 – 52.
3. *Сафонов К.В.* О возможности вычислительного распознавания контекстно-свободных языков // Вычислительные технологии. 2005. Т. 10. № 4. С. 91 – 98.
4. *Сафонов К.В.*, *Егорушкин О.И.* О синтаксическом анализе и проблеме В.М. Глушкова распознавания контекстно-свободных языков Хомского // Вестник ТГУ. Приложение. 2006. № 17. С. 63 – 66.

2008 Теоретические основы прикладной дискретной математики № 2(2)

DOI 10.17223/20710410/2/3

УДК 512.816+512.548.7

О КВАЗИГРУППАХ, ВОЗНИКАЮЩИХ ИЗ ФИЗИЧЕСКОЙ СТРУКТУРЫ РАНГА (2, 2)

Р.М. Мурадов, В.А. Кыров

Горно-Алтайский государственный университет

E-mail: kfizika@gasu.ru

В данной работе квазигруппа, задающая физическую структуру ранга (2, 2), изотопией переводится к квазигруппе с правой единицей и с тождественной операцией взятия правого обратного элемента.

***Ключевые слова*:** *квазигруппа*, *физическая структура ранга* (2, 2).

Напомним [1], что квазигруппа – это множество $(G, \bullet)$ с однозначной разрешимостью уравнений $A \bullet X = B$, $X \bullet A = C$ ($\forall A, B, C \in G$), относительно $C \in G$. Если существует элемент $E \in G$: $X \bullet E = X$, $\forall X \in G$, то он называется правой единицей. Если уравнение $X \bullet Y = E$ однозначно разрешимо относительно Y, то элемент Y называется правым обратным к элементу X. Ниже изучаются квазигруппы с правой единицей и с тождественной операцией взятия правого обратного элемента. В конце приводятся примеры таких квазигрупп. Квазигрупповая терминология приводится по монографии [1].

1. Квазигруппа с правой единицей и с тождественной операцией взятия правого обратного элемента

Пусть $(C, \bullet)$ – квазигруппа, φ – левая обратная операция, а ψ – правая обратная операция.

Теорема 1. Квазигруппа $(C, \bullet)$ изотопна квазигруппе $(C, \circ)$ с правой единицей E и тождественной операцией взятия правого обратного элемента.

Доказательство. Зафиксируем произвольный элемент $E \in C$. Разрешая в квазигруппе $(C, \bullet)$ уравнение $Z = X \bullet E$ относительно X, получаем $X = \varphi(Z, E) = \varphi_E(Z)$, тогда приходим к квазигруппе $(C, *)$, изотопной исходной с бинарной операцией $Z = X * Y = \varphi_E(X \bullet Y)$. В новой квазигруппе уравнение $E = X * Y = \varphi_E(X \bullet Y)$ разрешим справа: $Y = \psi(X, \varphi^{-1}{}_E(E)) = \psi_E(X)$. Построим еще одну квазигруппу $(C, \circ)$ с бинарной операцией $Z = X \circ Y = X * \psi_E(Y)$, изотопную квазигруппе $(C, *)$. Очевидно, построенная таким способом квазигруппа $(C, \circ)$ изотопна квазигруппе $(C, \bullet)$. Бинарные операции связаны с помощью изотопии $T'' = (1, \psi_E, \varphi_E)$, т.е. $X \circ Y = \varphi_E(X \bullet \psi_E(Y))$. Докажем, что E – правая единица в квазигруппе $(C, \circ)$. Для этого воспользуемся леммой

Лемма. $\psi_E(E) = E$.

Доказательство. Из определения отображений φ_E и ψ_E следует, что $\psi_E(E) = \psi(E, \varphi^{-1}{}_E(E)) = u$. По определению правой обратной операции имеем $E = \varphi_E(E \bullet u)$. Из определения левой обратной операции тогда получаем $E \bullet u = E \bullet E$. По определению квазигруппы $(C, \bullet)$ тогда имеем $u = E$. ■

Из леммы тогда получаем: $X \circ E = \varphi_E(X \bullet \psi_E(E)) = \varphi_E(X \bullet E) = X$. Докажем теперь, что правый обратный к элементу квазигруппы $(C, \circ)$ совпадает с ним самим. Рассмотрим для этого уравнение $E = X \circ Y$ и разрешим его относительно второго аргумента. Действительно, $E = X \circ Y = \varphi(X \bullet \psi(Y, \varphi^{-1}{}_E(E)), E)$, значит, $X \bullet \psi(Y, \varphi^{-1}{}_E(E)) = E \bullet E$, следовательно, $\psi(Y, \varphi^{-1}{}_E(E)) = \psi(X, E \bullet E)$. Из доказанной выше леммы вытекает равенство $\varphi^{-1}{}_E(E) = E \bullet E$. Значит, $Y = X$, т.е. правый обратный к элементу квазигруппы $(C, \circ)$ совпадает с ним самим. ■

Ниже будут приведены примеры на применение доказанной теоремы.

2. Физическая структура ранга (2,2)

Во второй половине XX века построена теория физических структур. Эта теория появилась из анализа фундаментальных законов физики. Так, в частности, всем известный второй закон Ньютона дает пример физической структуры ранга (2, 2). Позже была установлена связь этой теории с алгеброй, в частности с теорией квазигрупп. Дадим точную формулировку физической структуры ранга (2, 2) [2]. Рассмотрим три множества M, N, B и отображение f: $M \times N \to B$. Пусть выполняются следующие аксиомы:

A1. $\forall\ i \in \Omega_M,\ \forall\ b \in B,\ \exists!\ \alpha \in N$: $f(i\alpha) = b$, где $\Omega_M \subset M$.

A2. $\forall\ \alpha \in \Omega_N,\ \forall\ b \in \Omega_B, \exists!\ i \in M$: $f(i\alpha) = b$, где $\Omega_N \subset N$.

A3. $\forall\ \langle i_0, i_1 \rangle \in M \times \Omega_M,\ \forall\ \langle \alpha_0, \alpha_1 \rangle \in N \times \Omega_N$, существует связь

$$f_{00} = g(f_{01}, f_{11}, f_{10}),$$

где $f_{mn} = f(i_m \alpha_n)$, $m, n = 0, 1$.

Если задано метрическое отображение f: $M \times N \to B$ и выполняются аксиомы A1 – A3, то говорят, что задана физическая структура ранга (2, 2).

Построим два вспомогательных отображения: F_j: $N \to \Omega_B$, $j \in \Omega_M$, с явным видом $F_j(\alpha) = f(j\alpha)$ и F_γ: $M \to B$, $\gamma \in \Omega_N$: $F_\gamma(i) = f(i\gamma)$. Из аксиом А1 и А2 следует биективность построенных отображений, т.е. $N \sim \Omega_B$, $M \sim B$.

Из аксиом А1, А2 следует, что на множествах M, N, B определена частичная трехбазисная квазигруппа с операцией f. Согласно аксиоме А3, это специальная трехбазисная квазигруппа, которую будем называть *феноменологически симметричной* и обозначать (M, N, B, f). Точное определение можно найти в [4].

Введем обозначения $x = f(i\gamma)$, $y = f(j\alpha)$. Тогда метрическое отображение можно переписать в виде

$$f(i\alpha) = f(F^{-1}{}_\gamma(x), F^{-1}{}_j(y)) = \overline{f}\,(x,y).$$

Для построенного отображения $\overline{f} : B \times \Omega_B \to B$ все приведенные выше аксиомы выполняются, т.е. оно также задает физическую структуру ранга (2, 2). Другими словами, определена частичная феноменологически симметричная квазигруппа $(B, \overline{f}\,)$.

Теорема 2. Для частичной феноменологически симметричной квазигруппы $(B, \overline{f}\,)$ выполняется тождество, обобщающее ассоциативность

$$X(YZ) = (XY)Z,$$

где X, $Y \in B$, $Z \in \Omega_B$. Ограничение первого аргумента отображения $\overline{f} : B \times \Omega_B \to B$ на Ω_B дает групповую операцию в Ω_B, т. е. Ω_B является группой. Отображение $\overline{f} : B \times \Omega_B \to B$ задает тогда группу преобразований множества B с параметрической группой Ω_B.

Доказательство можно найти в работе [2]. ■

Как известно, каждая группа изотопна квазигруппе из некоторого класса. Далее к такой квазигруппе будет применена теорема 1. Эти квазигруппы возьмем из классификации физической структуры ранга (2, 2) для $B = C \subset R^4$.

3. 4-метрическая физическая структура ранга (2, 2)

На примере 4-метрической физической структуры ранга (2, 2), т. е. когда $C = B \subset R^4$, построим квазигруппы с правой единицей и тождественной операцией взятия правого обратного элемента. Явный вид квазигрупп дается по классификации 4-метрической физической структуры ранга (2, 2), которую приведем по работе [3]. Следует отметить, что до конца удалось довести исследование не всех квазигрупп.

Теорема 3. С точностью до изотопии бинарные операции квазигрупп, изотопных группам, т. е. в подмножестве $C \subset R^4$ задающие 4-метрическую физическую структуру ранга (2, 2), принимают вид

$$f^1 = (x+\xi)^2 \exp[\varepsilon(w+\theta)],\ f^2 = (y+\eta)^2 \exp[k(w+\theta)],\ f^3 = (z+\zeta)^2 \exp[l(w+\theta)],\ f^4 = w-\theta; \tag{1}$$

$$f^1 = [(x+\xi)^2 + (y+\eta)^2] \exp\left[-2k \operatorname{arctg} \frac{y+\eta}{x+\xi}\right],\ f^2 = 2 \operatorname{arctg} \frac{y+\eta}{x+\xi} + w + \theta,$$

$$f^3 = (z+\zeta)^2 \exp[l(w+\theta)],\ f^4 = w-\theta; \tag{2}$$

$$f^1 = (x+\xi)^2 \exp\left[-2k \frac{y+\eta}{x+\xi}\right],\ f^2 = 2\frac{y+\eta}{x+\xi} + w + \theta,\ f^3 = (z+\zeta)^2 \exp[\varepsilon(w+\theta)],\ f^4 = w-\theta; \tag{3}$$

$$f^1 = x+\xi,\ \ f^2 = 2\frac{y+\eta}{x+\xi} + w + \theta,\ f^3 = z+\zeta - \frac{(y+\eta)^2}{2(x+\xi)},\ f^4 = w-\theta, \tag{4}$$

$$f^1 = x+\xi,\ f^2 = 2\frac{y+\eta}{x+\xi} + w + \theta,\ f^3 = (x+\xi)\ln(z+\zeta+y+\eta+x+\xi) - y - \eta,\ f^4 = w-\theta; \tag{5}$$

$$f^1 = (x+\xi)^2 \exp\left[-2k \frac{y+\eta}{x+\xi}\right],\ f^2 = 2\frac{y+\eta}{x+\xi} + w + \theta,\ f^3 = k(y+\eta) - x - \xi - k^2(z+\zeta),\ f^4 = w-\theta; \tag{6}$$

$$f^1 = (x+\xi)^2 \exp\left[-2k \frac{y+\eta}{x+\xi}\right],\ f^2 = 2\frac{y+\eta}{x+\xi} + w + \theta,\ f^3 = 2\frac{z+\zeta}{x+\xi} - k\left(\frac{y+\eta}{x+\xi}\right)^2,\ f^4 = w-\theta; \tag{7}$$

$$f^1 = [x-\xi-\zeta(y-\eta)]e^{c\theta}, f^2 = (y-\eta)e^{\theta}, f^3 = (z-\zeta)e^{(c-1)\theta}, f^4 = w-\theta; \tag{8}$$

$$f^1 = (x+\xi)e^{z},\ f^2 = (x+\xi)e^{\zeta},\ f^3 = (y+\eta)e^{w},\ f^4 = (y+\eta)e^{\theta}; \tag{9}$$

$$f^1 = [(x+\xi)^2 + (y+\eta)]^2 \exp(z+\zeta),\ f^2 = 2 \operatorname{arctg} \frac{y+\eta}{x+\xi} + w + \theta,\ f^3 = z-\zeta,\ f^4 = w-\theta; \tag{10}$$

$$f^1 = (x+\xi)y\eta\,,\ f^2 = z + \frac{1}{(x+\xi)y^2}\,,\ f^3 = \zeta + \frac{1}{(x+\xi)\eta^2}\,,\ f^4 = w+\theta\,, \tag{11}$$

где k, l, c, q – произвольные постоянные, $\varepsilon = 0{,}1$.

Результатом применения теоремы 1 к теореме 3 является

Теорема 4. Квазигруппы (1) – (10) из теоремы 3 изотопны квазигруппам с правой единицей $E = (0, 0, 0, 0)$ и тождественной операцией взятия правого обратного элемента, а квазигруппа (11) – квазигруппе с правой единицей $E = (0, 1, 0, 0)$:

$$f^1 = (x-\xi)e^{\varepsilon\theta}\,,\ f^2 = (y-\eta)e^{k\theta}\,,\ f^3 = (z-\zeta)e^{l\theta}\,,\ f^4 = w-\theta\,; \tag{1'}$$

$$f^1 = [(x-\xi)\cos\theta - (y-\eta)\sin\theta]e^{k\theta}, f^2 = [(x-\xi)\sin\theta + (y-\eta)\cos\theta]e^{k\theta}, f^3 = (z-\zeta)e^{l\theta}, f^4 = w-\theta; \tag{2'}$$

$$f^1 = (x-\xi)e^{k\theta}\,,\ f^2 = [(x-\xi)\theta + y - \eta]e^{k\theta}\,,\ f^3 = (z-\zeta)e^{\theta}\,,\ f^4 = w-\theta\,; \tag{3'}$$

$$f^1 = x-\xi\,,\ f^2 = (x-\xi)\theta + y - \eta\,,\ f^2 = \frac{(x-\xi)\theta^2}{2} + (y-\eta)\theta + z - \zeta\,,\ f^4 = w-\theta\,; \tag{4'}$$

$$f^1 = x-\xi, f^2 = (x-\xi)\theta + y - \eta, f^3 = (x-\xi)(e^{\theta} - \theta - 1) + (y-\eta)(e^{\theta} - 1) + z - \zeta, f^4 = w-\theta; \tag{5'}$$

$$\left.\begin{gathered} f^1 = (x-\xi)e^{k\theta}, f^2 = [(x-\xi)\theta + y - \eta]e^{k\theta}, \\ f^3 = \frac{(x-\xi)(e^{k\theta}(k\theta-1)-1) + (y-\eta)(k(e^{k\theta}-1))}{k^2} + z - \zeta, f^4 = w-\theta; \end{gathered}\right\} \tag{6'}$$

$$f^1 = (x-\xi)e^{k\theta}, f^2 = [(x-\xi)\theta + y - \eta]e^{k\theta}, f^3 = \left(\frac{(x-\xi)k\theta^2}{2} + (y-\eta)k\theta + z - \zeta\right)e^{k\theta}, f^4 = w-\theta; \tag{7'}$$

$$f^1 = [x-\xi-\zeta(y-\eta)]e^{c\theta}, f^2 = (y-\eta)e^{\theta}, f^3 = (z-\zeta)e^{(c-1)\theta}, f^4 = w-\theta; \tag{8'}$$

$$f^1 = (x-\xi)e^{\zeta}\,,\ f^2 = (y-\eta)e^{\theta}\,,\ f^3 = z-\zeta\,,\ f^4 = w-\theta\,; \tag{9'}$$

$$f^1 = [(x-\xi)\cos\theta - (y-\eta)\sin\theta]e^{\zeta}, f^2 = [(x-\xi)\sin\theta + (y-\eta)\cos\theta]e^{\zeta}, f^3 = z-\zeta, f^4 = w-\theta; \tag{10'}$$

$$f^1 = \frac{(x-\xi)\eta^2}{1-(x-\xi)\eta^2\zeta}, f^2 = \frac{[1-(x-\xi)\eta^2\zeta]y}{\eta}, f^3 = \frac{y^2 z - \eta^2\zeta - (x-\xi)y^2\eta^2 z\zeta}{[1-(x-\xi)\eta^2\zeta]y^2}, f^4 = w-\theta. \tag{11'}$$

Доказательство теоремы проиллюстрируем на примере квазигруппы (9). Остальные случаи аналогичны. Для аналитических построений используем доказательство теоремы 1. Квазигрупповая операция в $(C, \bullet)$ обозначается $Z = X \bullet Y$, где $X = (x, y, z, w) \in C$, $Y = (\xi, \eta, \zeta, \theta) \in C$. Как сказано в формулировке теоремы, $E = (0, 0, 0, 0)$. По теореме 1 бинарная операция изотопной квазигруппы $(C, *)$ имеет вид

$$X * Y = \{(x+\xi)e^{\zeta}, (y+\eta)e^{\theta}, \ln e^{z-\zeta}, \ln e^{w-\theta}\}.$$

Отображение ψ_E в явном виде задается уравнениями $\xi = -x$, $\eta = -y$, $\zeta = z$, $\theta = w$. Осуществляя подстановку $\xi = -\xi$, $\eta = -\eta$, $\zeta = \zeta$, $\theta = \theta$ в предыдущих уравнениях, приходим к бинарной операции (9') квазигруппы $(C, \circ)$. ■

ЛИТЕРАТУРА

1. *Белоусов В.Д.* Основы теории квазигрупп и луп. М.: Наука, 1967.
2. *Симонов А.А.* Обобщенное матричное умножение как эквивалентное представление теории физических структур // Приложение к книге Кулакова Ю.И. «Теория физических структур». М., 2004.
3. *Кыров В.А.* Классификация четырехмерных физических структур ранга (2, 2) // Приложение к книге Михайличенко Г.Г., Мурадова Р.М. «Физические структуры как геометрии двух множеств». Горно-Алтайск, 2008.
4. *Кыров В.А.* Трехбазисные квазигруппы с обобщенным тождеством Уорда // Прикладная дискретная математика. 2008. № 1.

DOI 10.17223/20710410/2/4
УДК 519.7

ПОСТРОЕНИЕ НОРМАЛЬНЫХ ПЕРИОДИЧЕСКИХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ИЗ ЦИКЛИЧЕСКИ МИНИМАЛЬНЫХ ЧИСЕЛ

А.Г. Поздеев

Томский государственный университет

E-mail: anatoliy.pozdeev@vitasw.com

Предлагается алгоритм с вычислительной сложностью $O(2^n/n)$, позволяющий по задаваемым значениям параметра путем склеивания циклов, порожденных циклически минимальными числами, строить двоичные нормальные периодические последовательности порядка n так, что при разных значениях параметра с равной вероятностью строятся попарно неэквивалентные последовательности из множества большой мощности. В случае простого n указываются выражения для вычисления последней и размера параметра.

***Ключевые слова*:** *нормальные периодические последовательности*, *последовательности де Брейна*, *циклически минимальные числа.*

Нормальные периодические последовательности (НПП), они же – последовательности де Брейна, они же – нормальные рекуррентные последовательности [1], интересны для криптографии в качестве гамм или ключевых потоков в поточных шифрах. Для этого важно уметь порождать их с равной вероятностью из достаточно большого множества по случайно выбираемым значениям ключевого параметра (ключа). Судя по обзору методов построения НПП [1], к настоящему времени эта проблема еще не нашла удовлетворительного решения и требует дальнейшего исследования. В данной заметке демонстрируется подход к ее решению, основанный на построении НПП путем склеивания циклов, порожденных циклически минимальными числами. Изложению результатов предшествуют необходимые определения и сведения о работах других авторов в этом направлении.

Для любого натурального n элементы в множестве $\{0, 1\}^n$ рассматриваются как булевы векторы (слова) длиной n или как n-значные (двоичные) числа.

Последовательность $s = s_1s_2\ldots$, где $s_i \in \{0, 1\}$ для каждого $i \geq 1$, называется *периодической*, или *циклом*, если для некоторого натурального m и любого $i \geq 1$ в ней $s_{i+m} = s_i$; в этом случае пишут $s = <s_1s_2\ldots s_m>$, а наименьшие m с указанным свойством называют *периодом* последовательности s. Через $W_n(s)$ обозначается множество всех n-значных двоичных чисел, содержащихся в цикле $s = <s_1s_2\ldots s_m>$, а именно: $W_n(s) = = \{s_is_{i+1}\ldots s_{i+n-1}: i = 1, 2, \ldots, m\}$. Наименьшее n, при котором все числа $s_is_{i+1}\ldots s_{i+n-1}$ для $i = 1, 2, \ldots, m$ различны, называется *порядком* цикла s. Цикл порядка n и периода $m = 2^n$ называется *НПП порядка n*. Две НПП $< a >$ и $< b >$ называются *эквивалентными*, если a получается из b некоторым циклическим сдвигом.

Два цикла $a = <a_1a_2\ldots a_m>$ и $b = <b_1b_2\ldots b_r>$ называются *n-смежными*, если они не имеют общего n-значного числа, т.е. $W_n(a) \cap W_n(b) = \varnothing$, но имеют общее $(n-1)$-значное число, т.е. $W_{n-1}(a) \cap W_{n-1}(b) \neq \varnothing$. Результатом *склеивания n-смежных циклов a и b по общему числу* $a_ia_{i+1}\ldots a_{i+n-2} = b_jb_{j+1}\ldots b_{j+n-2}$ в $\{0, 1\}^{n-1}$ является цикл $c = <c_1c_2\ldots c_{m+r}> = <a_1a_2\ldots a_ia_{i+1}\ldots a_{i+n-2}b_{j+n-1}b_{j+n}\ldots b_rb_1b_2\ldots b_jb_{j+1}\ldots b_{j+n-2}a_{i+n-1}a_{i+n}\ldots a_m>$. По построению последнего, период последовательности c равен сумме периодов a и b, $W_n(a) \cup W_n(b) = W_n(c)$ и, как следствие, $W_{n-1}(a) \cup W_{n-1}(b) = W_{n-1}(c)$.

Множество циклов C называется *цикловым разбиением* множества $\{0, 1\}^n$, если различные циклы в нем не имеют общих n-значных чисел и любое n-значное число содержится в некотором цикле из C. Исходя из любого такого множества циклов, можно путем последовательного склеивания n-смежных циклов построить некоторую НПП порядка n. Выбирая разные исходные цикловые разбиения множества $\{0, 1\}^n$ и варьируя порядок выбора циклов и в них общих чисел для склеивания, можно таким способом построить любую из возможных НПП заданного порядка. Впервые этот метод построения НПП предложил А.Н. Радченко [2]. (Его полное изложение можно найти в [1].) Позднее операция склеивания смежных циклов возникала в работах других авторов, в частности в [3 – 6], где с ее помощью НПП заданного порядка строятся рекурсивно из НПП меньшего порядка – склеиванием НПП порядка n в НПП порядка $n+1$ для $n \geq 2$. В [7] представлен алгоритм со сложностью реализации $O(2^n)$ битовых операций, позволяющий таким образом порождать попарно неэквивалентные НПП заданного порядка n, выбирая их из некоторого множества мощности 2^{n-2} с равной вероятностью по значению параметра из множества $\{0, 1\}^{n-2}$.

Пусть далее для $a = a_1\ldots a_n \in \{0, 1\}^n$ и $i \in \{1, 2, \ldots, n\}$ через $a << i$ обозначается результат циклического сдвига числа a на i знаков влево: $(a << i) = a_{i+1}\ldots a_na_1\ldots a_i$. По определению $(a << n) = a$. Число $a \in \{0, 1\}^n$ называется *циклически минимальным*, если $a \leq (a << i)$ для каждого $i = 1, 2, ..., n-1$. Множество всех циклически минимальных чисел в $\{0, 1\}^n$ обозначается далее CM_n.

Ниже показывается, что множество $< CM_n >$ всех циклов $< a >$ для $a \in CM_n$ является цикловым разбиением множества $\{0, 1\}^n$, и предлагается алгоритм с вычислительной сложностью $O(2^n/n)$, позволяющий по значению параметра длиной $(|CM_n|-3)\lceil \log(n-1) \rceil$ бит путем склеивания циклов в $< CM_n >$ в определенном порядке построить некоторую НПП порядка n так, что при разных значениях параметра с равной вероятностью строятся попарно неэквивалентные НПП из множества, мощность которого в случае простого n равна $\prod_{m=1}^{n-1} m^{C_n^m / n}$. Длина параметра в этом случае аппроксимируется числом $2^n(\log n)/n$ бит.

Лемма 1. $\forall a, b \in CM_n\ (a \neq b \Rightarrow \neg\exists\, i \in \{1, 2, \ldots, n\}(a = (b << i)))$.

Доказательство. Предположив противное, будем иметь $b \le (b << i) = a$, $a \le (a << (n-i)) = b$ и $a = b$, что не так. ■

Следствие. $\forall a, b \in CM_n\ (a \neq b \Rightarrow W_n(< a >) \cap W_n(< b >) = \varnothing)$. ■

Лемма 2. $\forall a \in \{0, 1\}^n \exists\, j \in \{1, 2, \ldots, n\}((a << j) \in CM_n)$.

Доказательство. Определим $j \in \{1, 2, \ldots, n\}$ из условия: $(a << j)$ есть минимальное из чисел $(a << i)$ для всех $i \in \{1, 2, \ldots, n\}$. ■

Следствие. $\bigcup_{a \in CM_n} W_n(< a >) = \{0, 1\}^n$. ■

Объединяя утверждения следствий из лемм 1 и 2, получаем следующее

Утверждение 1. $< CM_n >$ есть цикловое разбиение множества $\{0, 1\}^n$. ■

Лемма 3. Пусть $i, j \in \{1, 2, \ldots, n\}$, $a = a_1 \ldots a_{i-1} 1 a_{i+1} \ldots a_n \in \{0, 1\}^n$ и $b = (a_1 a_2 \ldots a_{i-1} 0 a_{i+1} \ldots a_n << j)$. Тогда

1) циклы $< a >$ и $< b >$ n-смежные, и $a_{i+1} \ldots a_n a_1 a_2 \ldots a_{i-1}$ есть их общее $(n-1)$-значное число;

2) $\exists\, j \in \{1, 2, \ldots, n\}(b \in CM_n\ \&\ b < a)$.

Доказательство. 1) $a_{i+1} \ldots a_n a_1 a_2 \ldots a_{i-1} \in W_{n-1}(< a >) \cap W_{n-1}(< b >)$ и, кроме того, $W_n(< a >) \cap W_n(< b >) = \varnothing$, так как вес (число единиц) любого слова в $W_n(< a >)$ на 1 больше веса любого слова в $W_n(< b >)$;

2) по лемме 2 $\exists\, j \in \{1, 2, \ldots, n\}(b \in CM_n)$ и для такого j имеем $b \le (b << n - j) = a_1 a_2 \ldots a_{i-1} 0 a_{i+1} \ldots a_n < a$. ■

Обозначим $w(a)$ вес вектора $a \in \{0, 1\}^n$. Пусть $CM_n = \{r_0, r_1, \ldots, r_k\}$ и $r_0 < r_1 < \ldots < r_k$. Тогда $r_0 = 00\ldots0$, $r_1 = 0\ldots01$ и $r_k = 11\ldots1$. Для любого $t = 0, 1, \ldots, k$ определим r_t' из условий: $< r_t' > = < r_t >$ и длина r_t' есть период последовательности $< r_t >$, и положим $P_t = \{1, 2, \ldots, w(r_t')\}$. В частности, $r_0' = 0$, $r_k' = 1$, $r_1' = r_1$, $P_0 = \varnothing$, $P_1 = \{1\}$, $P_k = \{1\}$. Элементы в $P(n) = P_2 \times \ldots \times P_{k-1}$ будем называть *параметрами*.

Алгоритм *Z*. Выберем произвольный параметр $p = (p_2, \ldots, p_{k-1}) \in P(n)$, положим $p_1 = p_k = 1$ и построим последовательность $s(p)$ как результат последовательного склеивания циклов $< r_0 >, < r_1 >, \ldots, < r_k >$, выполненного по правилу: результат $< s >$ склеивания циклов $< r_0 >, < r_1 >, \ldots, < r_t >$ для каждого $t = 0, 1, \ldots, k-1$ склеивается с циклом $< r_{t+1} >$ по общему числу $\alpha(p_{t+1}) = a_{i+1} \ldots a_n a_1 a_2 \ldots a_{i-1}$, такому, что $r_{t+1} = a_1 \ldots a_{i-1} 1 a_{i+1} \ldots a_n$ и $p_{t+1} = w(a_1 \ldots a_{i-1} 1)$.

Корректность алгоритма *Z*. Нужно убедиться в том, что число $a_{i+1} \ldots a_n a_1 a_2 \ldots a_{i-1}$ действительно общее для циклов $< s >$ и $< r_{t+1} >$. По 2 в лемме 3 найдется $j \in \{1, 2, \ldots, n\}$, что $(a_1 \ldots a_{i-1} 0 a_{i+1} \ldots a_n << j) = b \in CM_n$, $b < r_{t+1}$ и, следовательно, $b = r_l$ для некоторого $l \in \{1, 2, \ldots, t\}$. По 1 в лемме 3 $a_{i+1} \ldots a_n a_1 a_2 \ldots a_{i-1}$ является общим числом циклов $< r_l >$ и $< r_{t+1} >$, поэтому $a_{i+1} \ldots a_n a_1 a_2 \ldots a_{i-1} \in \bigcup_{i=1}^{t} W_{n-1}(< r_i >) = W_{n-1}(< s >)$, и корректность Z доказана.

Ввиду утверждения 1 справедливо

Утверждение 2. Последовательность $s(p)$, порождаемая алгоритмом Z, есть НПП порядка n. ■

Пример. Пусть $n = 3$. Тогда $k = 3$, $r_0 = 000$, $r_1 = 001$, $r_2 = 011$, $r_3 = 111$, $r_0' = 0$, $r_1' = 001$, $r_2' = 011$, $r_3' = 1$, $P_2 = \{1, 2\}$, $P(3) = P_2 = \{1, 2\}$. Следуя алгоритму Z, можно построить две (и это будут все возможные для данного порядка неэквивалентные) НПП: $s(1)$ и $s(2)$. Построим первую. Имеем: $p = p_2 = 1$, $p_1 p_3 = 11$ и $< s > = < r_0 >$. Поскольку $r_1 = a_1 a_2 a_3 = 001$ и $p_1 = 1 = w(001)$, то $i = 3$ и $< s >$ и $< r_1 >$ склеиваются по общему числу $a_1 a_2 = 00$, давая результатом $< s > = <0001>$. Поскольку $r_2 = a_1 a_2 a_3 = 011$ и $p_2 = 1 = w(01)$, то $i = 2$ и $< s >$ и $< r_2 >$ склеиваются по общему числу $a_3 a_1 = 10$, давая результатом $< s > = <0001011>$. Наконец $< s >$ и $< r_3 >$ склеиваются по $a_2 a_3 = 11$, поскольку $r_3 = a_1 a_2 a_3 = 111$, $p_3 = 1 = w(1)$ и $i = 1$. Результатом этого склеивания является НПП $s(1) = <00010111>$. Аналогичным образом строится НПП $s(2)$, а именно: $< r_0 >$ и $< r_1 >$ склеиваются по числу 00 в цикл <0001>, который по 01 склеивается с $< r_2 >$ в цикл <0001101>, который по 11 склеивается с $< r_3 >$ в НПП $s(2) = <00011101>$.

Лемма 4. $p \neq p' \Rightarrow s(p) \neq s(p')$.

Доказательство. Пусть $p = p_2 \ \ldots\ p_{k-1}$ и $p = p'_2 \ \ldots\ p'_{k-1}$. Тогда, по построению в алгоритме Z, имеем $p_{t+1} \neq p'_{t+1} \Rightarrow \alpha(p_{t+1}) \neq \alpha(p'_{t+1}) \Rightarrow s(p) \neq s(p')$. ■

Утверждение 3. НПП, порождаемые алгоритмом Z с различными параметрами, не эквивалентны.

Доказательство. Поскольку НПП, порождаемые алгоритмом Z, начинаются с одного и того же слова – $00\ldots0$, то утверждение верно в силу леммы 4. ■

Лемма 5. Пусть n – простое число и $CM_n = \{r_0, r_1, \ldots, r_k\}$. Тогда

$$| CM_n| = (2^n–2)/n +2 \text{ и } \prod_{t=1}^{k-1} w(r_t) = \prod_{m=1}^{n-1} m^{C_n^m/n} . \quad (1)$$

Доказательство. В соответствии с утверждением 1 множество $\{0, 1\}^n$ при простом n разбивается на k+1 = $|CM_n|$ классов $C_0, C_1, \ldots, C_k$, где $C_0 = \{r_0\}$, $C_k = \{r_k\}$ и C_t для $t = 1, 2, \ldots, k–1$ состоит из n различных чисел ($r_t << j$), $j = 1, 2, ..., n$. Следовательно, для любого $m \in \{1, 2, \ldots, n–1\}$ среди чисел $r_1, \ldots, r_{k–1}$ будет ровно C_n^m/n чисел веса m, и произведение весов последних равно $m^{C_n^m/n}$, поэтому, во-первых, верно второе равенство в (1) и, во-вторых, $|CM_n| = 2 + \sum_{m=1}^{n-1} C_n^m / n = 2 + (2^n–2)/n$. ■

Утверждение 4. Количество всех НПП простого порядка n, порождаемых алгоритмом Z, равно

$$\prod_{m=1}^{n-1} m^{C_n^m/n} .$$

Доказательство. По лемме 4 количество всех НПП, порождаемых Z, равно $|P(n)|$. Пусть n простое. Тогда $r_t = r'_t$ для любого $r_t \in CM_n – \{r_0, r_k\}$ и, так как $w(r_1) =1$, то $|P(n)| = w(r_1)\cdot w(r_2)\cdot\ldots\cdot w(r_{k–1})$. Утверждение следует теперь из второго равенства в (1). ■

Утверждение 5. Для простого n длина параметра в $P(n)$ в битах равна $((2^n–2)/n –1) \lceil \log (n–1) \rceil$ и для большого n аппроксимируется числом $2^n(\log n)/n$.

Доказательство. По определению $P(n)$ длина $L(n)$ элемента $p = (p_2, \ldots, p_{k–1}) \in P(n)$ в битах равна $(k–2)\lceil \log(n–1) \rceil$, где k+1 = $|CM_n|$, поэтому $L(n) = (|CM_n|–3) \lceil \log(n–1) \rceil$. Подставляя сюда при простом n выражение для $|CM_n|$ из (1), получим требуемую формулу. ■

ЛИТЕРАТУРА

1. *Агибалов Г.П.* Нормальные рекуррентные последовательности // Вестник ТГУ. Приложение. 2007. № 23. С. 4 – 11.
2. *Радченко А.Н.* Методы синтеза кодовых колец // Радиотехника и электроника. 1959. № 11. С. 1782 – 1795.
3. *Lempel A.* On a Homomorphism of the De Bruijn Graph and Its Applications to the Design of Feedback Shift Registers // IEEE Trans. Computers. 1970. V. C-19. No. 12. P. 1204 – 1209.
4. *Mykkeltveit J.*, *Siu M.K.*, *Tong P.* On the cycle structure of some nonlinear shift register sequence // Information and Control. 1979. V. 43. P. 202 – 215.
5. *Siu M.K.*, *Tong P.* Generation of some de Bruijn sequences // Discrete Mathematics. 1980. V. 31. P. 97 – 100.
6. *Games R.A.* A generalized recursive construction for de Bruijn sequences // IEEE Trans. Inform. Theory. 1983. V. IT-29. No. 6. P. 843 – 850.
7. *Annexstein F.S.* Generating De Bruijn Sequences: An Efficient Implementation// IEEE Trans. Computers. 1997. V. C-46. No. 2. P. 198 – 200.

2008 Теоретические основы прикладной дискретной математики № 2(2)

DOI 10.17223/20710410/2/5

УДК 681.03

АЛГЕБРАИЧЕСКИЙ ИММУНИТЕТ БУЛЕВЫХ ФУНКЦИЙ

М.Э. Тужилин

Российский государственный гуманитарный университет, г. Москва

E-mail: mtmt@rambler.ru

Сделан обзор публикаций, посвященных построению функций с максимально возможным алгебраическим иммунитетом.

Ключевые слова: *алгебраический иммунитет*, *булевы функции*, *алгебраическая атака*, *криптография.*

В [1] был предложен новый метод криптографического анализа фильтрующего генератора. Рассмотрена следующая задача. Предполагается, что известна функция усложнения f и характеристический многочлен линейной рекуррентной последовательности (ЛРП) $h(x)$, определяющие фильтрующий генератор. Требуется по отрезку выходной последовательности длины m определить ключ схемы – начальное заполнение генератора – $k_0, k_1, \ldots, k_{n-1}$. Задача сводится к решению системы уравнений вида

$$\begin{cases} b_0 = f(k_0, k_1, ..., k_{n-1}), \\ b_1 = f(L(k_0, k_1, ..., k_{n-1})), \\ ... \\ b_{m-1} = f(L^{m-1}(k_0, k_1, ..., k_{n-1})), \end{cases} \quad (1)$$

где L – линейная форма, определяемая многочленом $h(x)$.

Можно рассматривать аналогичную задачу и для необязательно последовательных знаков выходной последовательности.

Суть предложенного авторами метода, который они назвали «алгебраической атакой», заключается в следующем. Произвольное уравнение из системы (1) можно записать в виде

$$f(s) = b_t. \quad (2)$$

Степень нелинейности этого уравнения определяется степенью нелинейности функции f. Если для специально подобранной булевой функции g выполняется условие $d = \deg fg < \deg f$, то, умножив обе части уравнения (2) на g, мы получим, например, для $b_t = 0$ уравнение

$$f(s)g(s) = 0,$$

которое имеет степень нелинейности d.

В результате, если для функции f существует функция g с минимально возможным значением параметра d, то исходная задача может быть сведена к решению переопределенной системы уравнений, для решения которой разработан ряд методов (линеаризация, вычисление базиса Грёбнера, XSL-метод и т.д.).

В работе [2] алгебраическая атака получила новое развитие и новое название – «быстрая алгебраическая атака». Этот метод основан на существовании тождеств низкой степени, связывающих значения функции f и значения её аргументов $k_0, k_1, \ldots, k_{n-1}$.

Позднее появились и другие работы, в которых предложенный метод был улучшен и обобщён, например [3 – 5]. В работе [5] было продемонстрировано на реально существующей криптосистеме, как работает алгебраическая атака.

Естественно, появились работы, в которых предпринимались попытки разработать теорию противостояния предложенным методам.

В работе [6] было введено следующее понятие алгебраического иммунитета. *Алгебраическим иммунитетом AI(f) булевой функции f* называется минимальное значение d, такое, что существует булева функция g степени d, не равная тождественно 0, аннулирующая функцию g или её отрицание, то есть выполнено соотношение $fg = 0$ или соотношение $(f + 1)g = 0$.

Очевидно, что чем выше алгебраический иммунитет функции, тем сложнее применение алгебраической атаки к фильтрующему генератору, использующему выходную функцию f.

1. Основные понятия и обозначения теории булевых функций

Введем необходимые обозначения и напомним основные понятия теории булевых функций.

Множество булевых функций от n переменных обозначается через B_n. Носитель $\mathrm{supp}(f)$ булевой функции f определяется как множество $\{(x_1, \ldots, x_n) \mid f(x_1, \ldots, x_n) = 1\}$. Под весом Хемминга $\mathrm{wt}(f)$ булевой функции f понимается мощность $|\mathrm{supp}(f)|$. Равновероятной (или сбалансированной) называется булева функция, удов-

летворяющая условию $\mathrm{wt}(f) = 2^{n-1}$. Расстоянием Хемминга $d(f, g)$ между двумя булевыми функциями f и g называется величина $\mathrm{wt}(f \oplus g)$.

Любая булева функция f может быть однозначно представлена многочленом Жегалкина (алгебраической нормальной формой – *ANF*(*f*)). Число переменных в терме наивысшего порядка с ненулевым коэффициентом называется алгебраической степенью функции f и обозначается через $\deg(f)$. Мерой «удаленности» от множества A_n аффинных функций (то есть функций, для которых выполнено условие $\deg(f) \le 1$) является степень нелинейности булевой функции $nl(f) = \min\limits_{g \in A_n} d(f, g)$. Этот параметр может быть вычислен с помощью разложения Фурье функции f. Если разложение Фурье имеет вид

$$F_f(a) = \sum_{x \in GF(2)^n} (-1)^{f(x)+(x,a)},$$

то

$$nl(f) = 2^{n-1} - \frac{1}{2}\max\left|F_f(a)\right|.$$

2. Основные результаты об алгебраическом иммунитете

В [1] (точнее, в расширенном варианте работы, выложенном в Интернете) доказано, что для любой функции f справедлива оценка

$$AI(f) \le \left\lceil \frac{n}{2} \right\rceil.$$

В [7] доказано, что доля равновероятных функций от n переменных, для которых выполнены неравенства

$$\frac{n}{2} - \sqrt{\frac{n}{2}\ln n} \le AI(f) \le \frac{n+1}{2},$$

стремится к 1 при $n \to \infty$.

Обозначим через $B_{n,k}$ множество булевых функций от n переменных, имеющих алгебраический иммунитет, равный k. В [8] показано, что для всех значений k, удовлетворяющих условию

$$k \le \left\lceil \frac{n}{2} \right\rceil,$$

множество $B_{n,k}$ не пусто.

Очевидно, что

$$|B_{n,0}| = 2.$$

В [9] доказано равенство

$$\left|B_{n,1}\right| = 2 - 2^{n+1} + \sum_{m=1}^{2^n-1}\sum_{r=1}^{n} F_n(m, r)\cdot 2^{r+1}\cdot(2^{2^n-r}-1)\cdot(-1)^{m+1},$$

где $F_n(m, r) = \dfrac{f_n(m, r)}{m!}$, а $f_n(m, r)$ задается итеративно:

$$f_n(m, r) = \begin{cases} 0, & \text{если } r > m, \\ f_n(m-1, r)\cdot(2^r - m) + f_n(m-1, r-1)\cdot(2^n - 2^{r-1}), & \text{если } r \le m. \end{cases}$$

3. Построение функций с максимально возможным алгебраическим иммунитетом

Ряд работ посвящён построению классов булевых функций, имеющих максимально возможный алгебраический иммунитет, равный $\left\lceil \frac{n}{2} \right\rceil$. Существует несколько подходов к построению таких функций. Они обсуждаются в [10, 11].

Первый из них заключается в итеративном конструировании булевых функций от $2k$ переменных с алгебраическим иммунитетом, равным k. Впервые он был опубликован в [12].

Обозначим через φ_{2k} булеву функцию от $2k$ переменных, определенную рекурсивно:

$$\varphi_{2k+2} = \varphi_{2k} \,||\, \varphi_{2k} \,||\, \varphi_{2k} \,||\, \varphi^1_{2k},$$

где символ || обозначает конкатенацию таблиц истинности, а φ^1_{2k} определяется рекурсивно:

$$\varphi^i_{2j} = \varphi^{i-1}_{2j-2} \,||\, \varphi^i_{2j-2} \,||\, \varphi^i_{2j-2} \,||\, \varphi^{i+1}_{2j-2} \text{ для } j > 0,\ i > 0$$

и начальными условиями $\varphi^0_j = \varphi_j$ для $j > 0$, $\varphi^0_i = i \bmod 2$ для $i \ge 0$.

Доказано, что $AI(\varphi_{2k}) = k$.

Свойства построенных функций изучаются в [13]. Во-первых, там показано, что эти функции не равновероятны. Во-вторых, их алгебраические степени близки к $2k$, но степени нелинейности таких функций равны

$$2^{n-1} - \binom{n-1}{\frac{n}{2}},$$

то есть далеки от максимума.

Второй подход основан на модифицировании симметрических функций [11, 14]. Введем в рассмотрение функции следующего вида:

$$f(x_1, \ldots, x_{2k}) = \begin{cases} 0, \text{ если } \mathrm{wt}(x_1, \ldots, x_{2k}) \le k, \\ 1 - \text{в противном случае.} \end{cases}$$

Доказано, что функции от $2k + 1$ переменных вида $x_{2k+1} + f(x_1, \ldots, x_{2k})$ имеют алгебраический иммунитет $k + 1$ и что они равновероятны. Такой же алгебраический иммунитет имеют функции вида $x_{2k+2} + x_{2k+1} + f(x_1, \ldots, x_{2k})$.

Этот же подход к построению булевых функций с оптимальным значением алгебраического иммунитета был независимо предложен в [15].

Булевы функции от нечетного числа переменных рассматриваются также в [16].

Из [17] известно, что булева функция от нечетного числа переменных с максимальным алгебраическим иммунитетом всегда равновероятна. Авторами предложен метод построения функций от нечетного числа переменных с максимальным алгебраическим иммунитетом и высокой степенью нелинейности.

Важный частный случай исследован в [18]. Там представлена рекурсивная процедура построения всех симметрических булевых функций от 2^m переменных, имеющих максимально возможный алгебраический иммунитет, равный 2^{m-1}. Доказано, что число таких функций равно $3 \cdot 2^m$.

В [19] для произвольной булевой функции предложен метод доказательства того, что она имеет фиксированный алгебраический иммунитет. На его основе получен способ построения равновероятных функций от нечетного числа переменных с максимально возможным алгебраическим иммунитетом.

Этот метод обобщён в [20], и там построен новый класс равновероятных функций с максимально возможным алгебраическим иммунитетом.

4. Алгоритмы вычисления алгебраического иммунитета

Отметим, что вычисление алгебраического иммунитета для произвольной булевой функции – довольно сложная вычислительная задача. Поэтому ряд работ посвящён проблемам построения эффективных алгоритмов вычисления алгебраического иммунитета для булевых функций, заданных различными способами (в виде многочлена Жегалкина, в виде ДНФ, с помощью функции «след» и т. д.). Выделим работу [21], посвященную этой проблематике.

В [22] предложено два алгоритма нахождения алгебраического иммунитета S-боксов, построенных на основе степенного отображения.

В [23] получена классификация всех равновероятных функций от пяти переменных по отношению к алгебраическому иммунитету. Всего таких функций 601080390. В таблице приведено количество функций с заданным значением $AI(f)$ и их доля среди всех рассмотренных функций.

$AI(f)$	1	2	3
Число равновероятных функций	62	403315208	197765120
Доля от общего количества	10^{-7}	0,671	0,329

5. Новые направления исследований

Дальнейшее развитие рассматриваемая тематика исследований получила в двух направлениях, которые объединяет понимание того факта, что максимальный алгебраический иммунитет функции является необходимым, но недостаточным условием для гарантирования её высоких криптографических качеств.

Первое направление можно проиллюстрировать следующим примером. Пусть функция f имеет максимально возможный алгебраический иммунитет, равный $\left\lceil \frac{n}{2} \right\rceil$. Могут найтись функции g и h, удовлетворяющие свойству $fg = h$, причём $\deg h = \left\lceil \frac{n}{2} \right\rceil$, но $\deg g < \left\lceil \frac{n}{2} \right\rceil$. В этом случае низкая степень функции g может быть использована для построения быстрой алгебраической атаки, несмотря на максимальное значение $AI(f)$.

В [24] построено семейство равновероятных функций от четного числа n неизвестных с алгебраическим иммунитетом $n/2$, таких, что для функций f и g, удовлетворяющих условиям $fg = h$ и $\deg h = n/2$, выполнено условие $\deg g \geq n/2$.

Второе направление исследований связано с нахождением для оптимальных с точки зрения алгебраического иммунитета функций их алгебраической степени и степени нелинейности. Ведется поиск функций, значения всех параметров которых близки к оптимальным.

В [7] обсуждается связь между алгебраическим иммунитетом и степенью нелинейности функции. Доказана следующая оценка для произвольной булевой функции f с $AI(f) = k$:

$$nl(f) \geq 2^{n-1} - \sum_{i=k-1}^{n-k} \binom{n-1}{i} = 2\sum_{i=0}^{k-2} \binom{n-1}{i}.$$

В качестве следствия из неё для функций с максимальным значением $AI(f) = \left\lceil \frac{n}{2} \right\rceil$ получены следующие оценки.

Если n нечетно, то $nl(f) \geq 2^{n-1} - \binom{n-1}{\frac{n-1}{2}}$, если n четно, то $nl(f) \geq 2^{n-1} - \binom{n-1}{\frac{n}{2}}$.

В [10] построено несколько классов равновероятных функций с оптимальным значением $AI(f)$ и со степенями нелинейности, равными

$$2^{n-1} - \binom{n-1}{\frac{n}{2}-1} + 2\binom{n-2}{\frac{n}{2}-2}/(n-2) \text{ для четного } n$$

и

$$2^{n-1} - \binom{n-1}{\frac{n-1}{2}} + \Delta(n) \text{ для нечетного } n.$$

При этом $\Delta(n) = \begin{cases} 2\sum_{i=0}^{k-1} \binom{3k-2}{k+i-1} \frac{k-i}{k}, & n = 4k+1, \quad k \geq 4, \\ 2\sum_{i=0}^{k+1} \binom{3k-1}{k+i} \frac{k+2-i}{k+2}, & n = 4k+3, \quad k \geq 5 \end{cases}$

и $\Delta(15) = 268$, $\Delta(19) = 2436$.

Обсуждается там и вопрос об алгебраической степени построенных функций.

В [25] рассмотрены обобщения понятия алгебраического иммунитета на случай k-значных функций, заданных на конечном поле, введены понятия алгебраического иммунитета для блочного шифра и для поточного шифра, приведены оценки исследуемых параметров, указана связь этих параметров с вычислением базисов Грёбнера для ряда мономиальных упорядочиваний.

6. Связь между алгебраическим иммунитетом и нелинейностью *r*-го порядка

В [26] введено понятие профиля нелинейности булевой функции, то есть последовательности, r-й член которой равен степени нелинейности r-го порядка $nl_r(f)$, равной расстоянию функции f до класса функций, чья алгебраическая степень не превосходит r. В [27] получена нижняя оценка для параметра $nl_r(f)$ для функции f с $AI(f) = k$:

$$nl_r(f) \geq \sum_{i=0}^{k-r-1} \binom{n}{i} + \sum_{i=k-2r}^{k-r-1} \binom{n-r}{i} \tag{3}$$

и обсуждаются вопросы, связанные с применением данной характеристики булевой функции к криптоанализу.

В [28] применен новый подход к получению нижних оценок параметра $nl_r(f)$. Доказана теорема, следствием из которой является оценка (3). Для случая $r = 2$ эта оценка улучшена. Доказано, что для функции f с $AI(f) = k$ выполнено неравенство

$$nl_2(f) \geq \sum_{i=0}^{k-1} \binom{n}{i} + \sum_{i=0}^{k-1} 2^i \binom{n-2i-1}{k-1-i}.$$

Более того, показано, что эта оценка достижима.

Заключение

Рассмотренная проблематика, посвященная алгебраическому иммунитету булевых функций, уже выросла из первоначально возникшей задачи противодействия «алгебраической атаке». По-существу, на её базе возникла новая область исследования булевых функций. В последних публикациях на эту тему широко применяются алгебраические и комбинаторные подходы к получению оценок рассматриваемых параметров булевых функций, использующие характеры конечного поля, коды Рида – Малера и другие.

Постоянно растущее число публикаций и появление новых имен исследователей свидетельствуют о растущем интересе к данной проблематике и ее актуальности.

ЛИТЕРАТУРА

1. *Courtois N.*, *Meier W.* Algebraic Attacks on Stream Ciphers with Linear Feedback // Proceedings of Eurocrypt 2003, Lecture Notes in Computer Sciences. 2003. V. 2656. P. 345 – 359.
2. *Courtois N.* Fast Algebraic Attacks on Stream Ciphers with Linear Feedback // Proceedings of Crypto 2003, Lecture Notes in Computer Sciences. 2003. V. 2729. P. 176 – 194.
3. *Armknecht F.* Improving Fast Algebraic Attacks // Proceedings of FSE 2004, Lecture Notes in Computer Sciences. 2004 V. 3017. P. 65 – 82.
4. *Hawkes P.*, *Rose G.* Rewriting Variables: the Complexity of Fast Algebraic Attacks on Stream Ciphers // Proceedings of Crypto 2004, Lecture Notes in Computer Sciences. 2004. V. 3152. P. 390 – 406.
5. *Courtois N.* Cryptanalysis of SFINKS // Proceedings of ISICS 2005, Lecture Notes in Computer Sciences. 2005. V. 3935. P. 261 – 269.
6. *Meier W.*, *Pasalic E.*, *Carlet C.* Algebraic Attacks and Decomposition of Boolean Functions // Proceedings of Eurocrypt 2004, Lecture Notes in Computer Sciences. 2004. V. 3027. P. 474 – 491.
7. *Didier F.* A new upper bound of the block error probability after decoding over the erasure channel // IEEE Transactions On Information Theory. 2006. V. 52. No. 10. P. 4496 – 4503.
8. *Lobanov M.* Tight bound between nonlinearity and algebraic immunity // Cryptology ePrint Archive 2005/441.
9. *Tu Z.*, *Yingpu Deng Y.* Algebraic Immunity Hierarchy of Boolean Functions // Cryptology ePrint Archive 2007/259.
10. *Carlet C.*, *Zeng X.*, *Li C.*, *Hu L.* Further properties of several classes of Boolean functions with optimal algebraic immunity // Cryptology ePrint Archive 2007/370.
11. *Dalai D.*, *Maitra S.*, *Sarkar S.* Basic theory in construction of Boolean functions with maximum possible annihilator immunity // Designs, Codes and Cryptography. 2006. V. 40. No 1. P. 41 – 58.
12. *Dalai D.*, *Gupta K.*, *Maitra S.* Cryptographically significant Boolean functions: construction and analysis in terms of algebraic immunity // Proceedings of FSE 2005, Lecture Notes in Computer Sciences. 2005. V. 3557. P. 98 – 111.
13. *Carlet C.*, *Dalai D.*, *Gupta K.*, *Maitra S.* Algebraic immunity for cryptographically significant Boolean functions: analysis and construction // IEEE Transactions on Information Theory. 2006. V. 52. No. 7. P. 3105 – 3121.
14. *Braeken A.*, *Preneel B.* On the algebraic immunity of symmetric Boolean functions // Proceedings of Indocrypt 2005, Lecture Notes in Computer Sciences. 2005. V. 3797. P. 35 – 48.
15. *Armknecht F.*, *Krause M.* Constracting Single- and Multi-output Boolean Functions with Maximal Algebraic Immunity // Proceedings of ICALP 2006, Lecture Notes in Computer Sciences. 2006. V. 4052. P. 180 – 191.
16. *Li N.*, *Qi W.* Construction and analysis of Boolean functions of $2t+1$ variables with maximum algebraic immunity // Proceedings of Asiacrypt 2006, Lecture Notes in Computer Sciences. 2006. V. 4284. P. 84 – 98.
17. *Dalai D.*, *Gupta K.*, *Maitra S.* Results on algebraic immunity for cryptographically significant Boolean functions // Proceedings of Indocrypt 2004, Lecture Notes in Computer Sciences. 2004. V. 3348. P. 92 – 106.
18. *Liu F.*, *Feng K.* On the 2^m-variable Symmetric Boolean Functions with Maximum Algebraic Immunity 2^{m-1} // Proceedings of WCC. 2007. P. 225 – 232.
19. *Carlet C.* A method of construction of balanced functions with optimum algebraic immunity // Cryptology ePrint Archive 2006/149.
20. *Wang Y.*, *Fan S.*, *Han W.* New construction of Boolean function with optimum algebraic immunity // Cryptology ePrint Archive 2008/176.
21. *Баев В.В.* Эффективные алгоритмы получения оценок алгебраической иммунности булевых функций // Дис. ... канд. физ.-мат. наук. М.: МГУ им. М.В.Ломоносова, 2007.
22. *Nawaz Y.*, *Gupta K.*, *Gong G.* Efficient Techniques to Find Algebraic Immunity of S-boxes Based on Power Mappings // Proceedings of WCC. 2007. P. 237 – 246.
23. *Canteaut A.* Open problems related to algebraic attacks on stream ciphers // Proceedings of WCC 2005, Lecture Notes in Computer Science. 2006. V. 3969. P. 120 – 134.
24. *Dalai D.*, *Maitra S.* Balanced Boolean Functions with (more than) Maximum Algebraic Immunity // Proceedings of WCC. 2007. P. 99 – 108.
25. *Ars G.*, *Faugère J.* Algebraic Immunities of functions over finite fields // INRIA, Rapport de recherché №5532. 2005.
26. *Carlet C.* On the higher order nonlinearities of algebraic immune Boolean functions // Proceedings of Crypto 2006, Lecture Notes in Computer Science. 2006. V. 4117. P. 584 – 601.
27. *Mesnager S.* Improving the lower bound on the higher order nonlinearity of Boolean functions with prescribed algebraic immunity // Cryptology ePrint Archive 2007/117.
28. *Lobanov M.* Tight bounds between algebraic immunity and nonlinearities of high orders // Cryptology ePrint Archive 2007/444.

2008 Теоретические основы прикладной дискретной математики № 2(2)
DOI 10.17223/20710410/2/6
УДК 519.7

О С-ШИРИНЕ КОНЕЧНЫХ АЦИКЛИЧЕСКИХ ГРУПП

В.М. Фомичев

Московский инженерно-физический институт (*государственный университет*)

E-mail: fomichev@nm.ru

Даны начальные результаты исследования представления конечной группы в виде покрытия максимальными циклическими подгруппами. Указан способ построения групп линейных подстановок множества V_n, у которых c-ширина не меньше 2^n.

***Ключевые слова*:** *циклическая группа*, *с-ширина группы.*

Конечная группа G имеет единственное несократимое представление в виде канонического c-покрытия, то есть в виде объединения максимальных циклических подгрупп [1,3]:

$$G = \bigcup_{g \in B_G} \langle g \rangle,$$

где B_G есть c-базис группы G, представляющий собой систему элементов группы G, порождающих все максимальные в G циклические подгруппы. Порядок множества B_G называется c-шириной группы G (обозначается $h_c(G)$).

Величина c-ширины группы G характеризует сложность определения наследственного признака H в группе G в рамках подхода [4, 5], предполагающего определение признака H во всех максимальных циклических подгруппах группы G.

Для c-ширины произвольной ациклической группы G верны оценки:

$$1 < h_c(G) \le |G| - 1, \tag{1}$$

где $h_c(G) = 1$ тогда и только тогда, когда G – циклическая группа. Верхняя оценка (1) достигается, в частности, когда G – прямая сумма нескольких циклических групп порядка 2. Если Σ_r – группа сдвигов пространства V_r двоичных r-мерных векторов, то $h_c(\Sigma_r)=2^r - 1$, так как c-базис группы Σ_r образуют все ненулевые векторы пространства V_r.

1. Свойства канонического *c*-покрытия конечной группы

Обозначим через N множество натуральных чисел. Далее считаем, что G – конечная ациклическая группа и её каноническое c-покрытие имеет вид

$$G = \bigcup_{i=1}^{h} \langle g_i \rangle, \tag{2}$$

где $\{g_1, \ldots, g_h\}$ есть c-базис группы G и $h = h_c(G) > 1$.

Обозначим через $\mathrm{Gen}\langle g \rangle$ множество элементов, порождающих циклическую подгруппу $\langle g \rangle$ группы G. Известно [2], что если $\mathrm{ord}\, g = n$, то $\mathrm{Gen}\langle g \rangle$ состоит из всех элементов вида g^t, где $(t, n) = 1$.

Утверждение 1. Если $(g_i)^t \in \mathrm{Gen}\langle g_i \rangle$, то при любом $\tau = 1, \ldots, \mathrm{ord} g_j$ элементы $(g_i)^t(g_j)^\tau$ и $(g_j)^\tau(g_i)^t$ группы G не принадлежат группе $\langle g_j \rangle$, где $i, j \in \{1, \ldots, h\}$ и $i \ne j$.

Доказательство. Если $(g_i)^t(g_j)^\tau \in \langle g_j \rangle$, то $(g_i)^t = (g_j)^k$ при некотором натуральном k. Следовательно, $\langle g_i \rangle$ – подгруппа группы $\langle g_j \rangle$, что невозможно в силу максимальности циклической подгруппы $\langle g_i \rangle$ в группе G. Для элемента $(g_j)^\tau(g_i)^t$ утверждение доказывается аналогично.

Следствие 1. Если $(g_i)^t \in \mathrm{Gen}\langle g_i \rangle$ и $(g_j)^\tau \in \mathrm{Gen}\langle g_j \rangle$, то элементы $(g_i)^t(g_j)^\tau$ и $(g_j)^\tau(g_i)^t$ группы G не принадлежат множеству $\langle g_i \rangle \cup \langle g_j \rangle$, где $i, j \in \{1, \ldots, h\}$ и $i \ne j$.

Теперь нижнюю из оценок (1) можно уточнить для ациклических групп.

Следствие 2. $h_c(G) > 2$.

Доказательство. В силу ацикличности группы G в ней имеется не менее двух максимальных циклических подгрупп. Пусть эти подгруппы порождаются элементами g_1 и g_2 соответственно. По следствию 1 утверждения 1 элемент $g_1 g_2 \notin \langle g_1 \rangle \cup \langle g_2 \rangle$. Следовательно, $h_c(G) > 2$.

Оценим c-ширину группы G через её порядок и порядки элементов её c-базиса. Обозначим: $\omega(G) = \max\limits_{g \in G} \mathrm{ord} g$. Заметим, что $\omega(G)$ совпадает с наибольшим из порядков максимальных циклических подгрупп группы G.

Утверждение 2. Если G – конечная ациклическая группа, то $h_c(G) \geq \dfrac{|G|-1}{\omega(G)-1}$.

Доказательство. Из равенства (2) следует, что $G\setminus\{e\} = \bigcup\limits_{i=1}^{h}(\langle g_i\rangle \setminus \{e\})$, так как единичный элемент e группы G является единичным элементом любой её подгруппы. Отсюда

$$|G| - 1 \leq \sum_{i=1}^{h} (\mathrm{ord} g_i - 1). \tag{3}$$

По определению $\mathrm{ord}\, g_i \leq \omega(G)$, $i = 1, \ldots, h$, поэтому из (3) получаем

$$|G| - 1 \leq h(\omega(G) - 1),$$

откуда следует требуемое неравенство.

Для циклической группы $\langle g\rangle$ порядка $n > 1$ назовём Gen-комплексом всякое её подмножество X порядка $r > 1$, удовлетворяющее условиям: $X \cap \mathrm{Gen}\langle g\rangle \neq \varnothing$ и $g^{t-\tau} \in \mathrm{Gen}\langle g\rangle$ для всех g^t, g^τ из X при $t > \tau$. Заметим, что при $g \neq e$ пара элементов $\{e, g\}$ является Gen-комплексом группы $\langle g\rangle$. Обозначим через $\mu(n)$ наибольший из порядков всех Gen-комплексов циклической группы порядка $n > 1$.

Утверждение 3. Величина $\mu(n)$ равна наименьшему простому делителю числа n.

Доказательство. Пусть p – наименьший простой делитель числа n, где $n > 1$. Тогда множество $\{e, g, \ldots, g^{p-1}\}$ содержит элемент g, порождающий группу $\langle g\rangle$ и $g^{t-\tau} \in \mathrm{Gen}\langle g\rangle$ для всех t, τ из $\{0, 1, \ldots, p-1\}$ при $t \neq \tau$, так как $(t - \tau, p) = 1$ в силу простоты числа p. Отсюда, раз натуральное число $t - \tau$ взаимно просто с p и меньше n, то $t - \tau$ взаимно просто и с любым другим делителем числа n. Следовательно, $(t - \tau, n) = 1$. Значит, множество $\{e, g, \ldots, g^{p-1}\}$ является Gen-комплексом группы $\langle g\rangle$.

Вместе с тем, если подмножество X (порядка $r > 1$) группы $\langle g\rangle$ содержит элементы g^t и g^τ, где $t \equiv \tau \pmod p$, то X не является Gen-комплексом группы $\langle g\rangle$, так как $g^{t-\tau} \notin \mathrm{Gen}\langle g\rangle$. Действительно, в этом случае $(t - \tau, p) = p$, откуда получаем, что $(t - \tau, n) \geq p > 1$. Следовательно, Gen-комплекс группы $\langle g\rangle$ содержит не более p элементов. Таким образом, $\mu(n) = p$.

Gen-комплекс $\{e, g, \ldots, g^{p-1}\}$ нетривиальной группы $\langle g\rangle$ порядка n, где p – наименьший простой делитель числа n, обозначим через $\mathrm{Gen}^*\langle g\rangle$. Установим соотношения между характеристиками элементов c-базиса ациклической группы G порядка n, определяемой представлением (2). Так как $h_c(G) \geq 3$, то число n – не простое. Следовательно, корректными являются следующие обозначения: $\mu_I = \mu(\mathrm{ord} g_i)$, $\eta_I = \max\limits_{j\in\{1,\ldots,h\}\setminus\{i\}} \mu_j$, $i = 1, \ldots, h$.

Теорема 1. Для ациклической группы G, определяемой представлением (2), при любом $i = 1, \ldots, h$ выполнено

$$|G| \geq \mathrm{ord} g_i \cdot \eta_i.$$

Если $\langle g_i\rangle \cap \langle g_j\rangle = \{e\}$ для $i, j \in \{1, \ldots, h\}$, $i \neq j$, то $|G| \geq \mathrm{ord} g_i \cdot \mathrm{ord} g_j$.

Доказательство. Пусть M – подмножество элементов группы G вида

$$M = \{(g_i)^t \cdot (g_j)^\tau\},$$

где $t = 0, 1, \ldots, \mathrm{ord}\, g_i - 1$, $\tau = 0, 1, \ldots, \theta - 1$ при некотором натуральном θ.

Так как $|G| \geq |M|$, то для доказательства теоремы достаточно показать, что множество M не содержит одинаковых элементов группы G при любом из двух условий: *а*) $\theta = \mu_j$; *б*) $\theta = \mathrm{ord}\, g_j$, если $\langle g_i\rangle \cap \langle g_j\rangle = \{e\}$. Предположим противное – пусть

$$(g_i)^t \cdot (g_j)^\tau = (g_i)^v \cdot (g_j)^w \tag{4}$$

при $t, v \in \{0, 1, \ldots, \mathrm{ord}\, g_i - 1\}$ и $\tau, w \in \{0, 1, \ldots, \theta - 1\}$, где $(t, \tau) \neq (v, w)$, тогда равенство (4) равносильно следующему равенству:

$$(g_i)^{t-v} = (g_j)^{w-\tau}. \tag{5}$$

В случае (*а*) $\langle (g_j)^{w-\tau}\rangle = \langle g_j\rangle$ при $w \neq \tau$, так как по утверждению 3 $(g_j)^w$, $(g_j)^\tau \in \mathrm{Gen}^*\langle g_j\rangle$ при $\theta = \mu_j$. Отсюда и из (5) следует включение $\langle g_j\rangle \subseteq \langle g_i\rangle$, противоречащее равенству (2) в силу максимальности подгруппы $\langle g_j\rangle$ в группе G. Следовательно, в случае (*а*) множество M не содержит одинаковых элементов группы G.

Рассмотрим случай (*б*). Если $t = v$ при $w \neq \tau$, то из (5) получаем равенство $(g_j)^{w-\tau} = e$, которое невозможно при любых различных $\tau, w \in \{0, 1, \ldots, \mathrm{ord}\, g_j - 1\}$.

Если $t \neq v$ при $w = \tau$, то из (5) получаем равенство $(g_i)^{t-v} = e$, которое невозможно при любых различных $t, v \in \{0, 1, \ldots, \mathrm{ord}\, g_i - 1\}$.

Если $t \neq v$ и $w \neq \tau$, то элементы $(g_i)^{t-v}$ и $(g_j)^{w-\tau}$ отличны от e и из (5) следует, что оба они принадлежат $\langle g_i\rangle \cap \langle g_j\rangle$, что противоречит условию случая (*б*).

Следовательно, и в случае (*б*) множество M не содержит одинаковых элементов группы G.

Следствие 1. Если $\mathrm{ord} g_i = \omega(G)$, то $h_c(G) > \eta_i$.

Доказательство. По теореме 1 $|G| \geq \mathrm{ord}\, g_i \cdot \eta_i$ для ациклической группы G при любом $i = 1, \ldots, h$. Отсюда получаем равносильное неравенство:

$$|G| - 1 \geq \mathrm{ord} g_i \cdot \eta_i - \eta_i + \eta_i - 1.$$

По определению $\eta_i > 1$ при любом $i = 1, \ldots, h$, поэтому из последнего неравенства следует:

$$|G| - 1 > (\mathrm{ord}\, g_i - 1)\eta_i.$$

Разделив обе части неравенства на натуральное число $\mathrm{ord}\, g_i - 1$, имеем

$$\frac{|G|-1}{\mathrm{ord} g_i - 1} > \eta_i,\ i = 1, \ldots, h.$$

В частности, при $\mathrm{ord}\, g_i = \omega(G)$ по утверждению 2 имеем, что $h_c(G) > \eta_i$.

Следствие 2. Если при некоторых $i, j \in \{1, \ldots, h\}$, где $i \neq j$, $\mathrm{ord}\, g_j$ – простое число и $\mathrm{ord}\, g_i \geq \mathrm{ord}\, g_j$, то $h_c(G) > \mathrm{ord}\, g_j$.

Доказательство. Пусть $\omega(G) = \mathrm{ord}\, g_k$, где $k \in \{1, \ldots, h\} \setminus \{j\}$, при указанном j такое k найдётся, так как $\mathrm{ord} g_i \geq \mathrm{ord} g_j$. Тогда по следствию 1 теоремы 1 $h_c(G) > \eta_k$, где $\eta_k \geq \mathrm{ord}\, g_j$ при простом $\mathrm{ord}\, g_j$.

Таким образом, c-ширина любой конечной ациклической группы G не меньше 3. Более точная нижняя оценка может быть получена с помощью числовых характеристик, определяемых порядками элементов c-базиса группы G.

2. Строение конечных ациклических групп c-ширины 3

Опишем строение ациклических групп, c-ширина которых равна 3.

Теорема 2. Если каноническое c-покрытие группы G есть $\langle g_1\rangle \cup \langle g_2\rangle \cup \langle g_3\rangle$, то $|G| = 4m$ при некотором $m \in N$, $\mathrm{ord} g_1 = \mathrm{ord} g_2 = \mathrm{ord} g_3 = 2m$ и $\langle (g_1)^2\rangle = \langle (g_2)^2\rangle = \langle (g_3)^2\rangle$. Для любого $m \in N$ имеется ациклическая группа G порядка $4m$, c-ширина которой равна 3.

Доказательство. 1) Пусть порядок одной из максимальных циклических подгрупп группы G равен 2. Например, $\mathrm{ord} g_1 = 2$. Тогда $\langle g_1\rangle \setminus \{e\} = \{g_1\}$ и по следствию 1 утверждения 1 получаем, что $g_2 g_3 = g_1 = g_3 g_2$. Докажем, что в этом случае $\mathrm{ord} g_2 = \mathrm{ord} g_3 = 2$.

Действительно, $g_2, (g_2)^{-1} \in \mathrm{Gen}\, \langle g_2\rangle$, откуда по следствию 1 утверждения 1 получаем, что $g_2 g_3 = g_1$ и $(g_2)^{-1} g_3 = g_1$. Следовательно, $g_2 = (g_2)^{-1}$ и $\mathrm{ord} g_2 = 2$. Равенство $\mathrm{ord} g_3 = 2$ доказывается аналогично.

Следовательно, если порядок одной из максимальных циклических подгрупп группы G равен 2, то теорема верна ($m = 1$).

2) Пусть $\min\{\mathrm{ord}\, g_1, \mathrm{ord}\, g_2, \mathrm{ord}\, g_3\} > 2$. Тогда $g_1 \neq (g_1)^{-1}$ и $(g_1)^2 \neq e$.

Обозначим $a = g_1 g_2$, $b = (g_1)^{-1} g_2$. Так как $g_1, (g_1)^{-1} \in \mathrm{Gen}\langle g_1\rangle$, то по следствию 1 утверждения 2 $a, b \in \langle g_3\rangle \setminus \{e\}$. Следовательно, $b^{-1} \in \langle g_3\rangle \setminus \{e\}$ и $ab^{-1} \in \langle g_3\rangle \setminus \{e\}$, где $ab^{-1} = (g_1)^2$.

Отсюда получаем, что

$$\langle (g_1)^2\rangle < \langle g_3\rangle, \tag{6}$$

где $\mathrm{ord}\, g_1$ – чётное число. Действительно, в случае нечётности числа $\mathrm{ord}\, g_1$ имеем $(g_1)^2 \in \mathrm{Gen}\langle g_1\rangle$, и, следовательно, $\langle g_1\rangle < \langle g_3\rangle$, что противоречит максимальности циклической подгруппы $\langle g_1\rangle$ в группе G. Заметим, что группа $\langle g_3\rangle$ не содержит нечётных степеней элемента g_1, так как в противном случае из включения (6) следовало бы включение $\langle g_1\rangle < \langle g_3\rangle$.

Аналогично доказываются другие включения типа (6), а именно: $\langle (g_i)^2\rangle < \langle g_j\rangle$, где группа $\langle g_j\rangle$ не содержит нечётных степеней элемента g_i, $i, j \in \{1, 2, 3\}$, $i \neq j$. Совокупность этих условий означает, что $\langle (g_i)^2\rangle < \langle (g_j)^2\rangle$, $i, j \in \{1, 2, 3\}$, $i \neq j$. Следовательно, $\langle (g_1)^2\rangle = \langle (g_2)^2\rangle = \langle (g_3)^2\rangle$ и множества $\langle g_1\rangle \setminus \langle (g_1)^2\rangle$, $\langle g_2\rangle \setminus \langle (g_2)^2\rangle$, $\langle g_3\rangle \setminus \langle (g_3)^2\rangle$ попарно не пересекаются. Таким образом, из канонического c-покрытия группы G вытекает разбиение группы G:

$$G = \langle (g_1)^2\rangle \cup (\langle g_1\rangle \setminus \langle (g_1)^2\rangle) \cup (\langle g_2\rangle \setminus \langle (g_2)^2\rangle) \cup (\langle g_3\rangle \setminus \langle (g_3)^2\rangle).$$

Так как $\mathrm{ord}\, g_i$ – чётное число (пусть оно равно $2m$ при некотором $m \in N$), то $|\langle g_i\rangle \setminus \langle (g_i)^2\rangle| = \mathrm{ord}\, \langle (g_i)^2\rangle = m$, $i = 1, 2, 3$. Следовательно, в указанном разбиении каждый блок имеет мощность m, откуда получаем, что $|G| = 4m$.

Существование для любого $m \in N$ ациклической группы G порядка $4m$, c-ширина которой равна 3, обеспечивается следующим построением. Пусть $G = \langle g_1, g_2, g_3\rangle$, где g_1, g_2, g_3 – попарно различные элементы порядка $2m$, $(g_1)^2 = (g_2)^2 = (g_3)^2$ и при любой перестановке (i, j, k) номеров 1, 2, 3 выполнено равенство множеств

$$\langle (g_i)^2\rangle \cdot g_j = \langle g_k\rangle \setminus \langle (g_k)^2\rangle.$$

Следствие 1. Если порядок группы G не делится на 4, то $h_c(G) > 3$.

Следствие 2. Если $|G| = 4$, то $h_c(G) = 3$ и группа G изоморфна группе сдвигов Σ_2.

Доказательство. Если группа G ациклическая и $|G| = 4$, то G состоит из нейтрального элемента и трех элементов порядка 2. Изоморфизм φ задается биекцией между элементами порядка 2 группы G и ненулевыми векторами группы Σ_2.

3. *c*-Ширина ациклических групп порядка *pq* при простых *p* и *q*

Пусть $|G| = pq$, где p и q – простые числа и $p \geq q \geq 2$. Заметим, что если $p = q = 2$, то $h_c(G) = 3$ по следствию 2 теоремы 2. В противном случае $h_c(G) > 3$ по следствию 1 теоремы 2. Уточним оценку величины $h_c(G)$ при $p + q > 4$.

Лемма 1. Пусть c-базис группы G состоит из n_1 элементов порядка p_1, …, n_k элементов порядка p_k, где $k \in N$ и p_1, …, p_k – простые числа. Тогда

$$h_c(G) = n_1 + \ldots + n_k,$$

$$|G| = 1 + n_1(p_1 - 1) + \ldots + n_k(p_k - 1).$$

Доказательство. Первое равенство следует из определения чисел n_1, …, n_k.

Для доказательства второго равенства заметим, что каноническое c-покрытие (2) группы G есть объединение $n_1 + \ldots + n_k$ циклических групп порядков p_1, …, p_k, где в силу простоты чисел p_1, …, p_k пересечение любых двух циклических групп из канонического c-покрытия состоит лишь из нейтрального элемента. Следовательно, множество $G\backslash\{e\}$ разбивается на $n_1 + \ldots + n_k$ блоков, мощности которых есть мощности циклических групп из канонического c-покрытия за вычетом нейтрального элемента.

Утверждение 4. Пусть $|G| = p^2$, тогда $h_c(G) = p + 1$.

Доказательство. В условиях утверждения группа G состоит из нейтрального элемента и элементов порядка p. Каноническое c-покрытие (2) группы G есть объединение $h_c(G)$ циклических групп порядка p. Отсюда по лемме 1 $|G| = 1 + h_c(G)(p - 1)$. Следовательно, $h_c(G) = \dfrac{p^2 - 1}{p - 1} = p + 1$.

Теорема 3. Пусть $|G| = pq$, где $p > q \geq 2$, тогда

1) $q + 1 \leq h_c(G) \leq p + \dfrac{p-1}{q-1}$;

2) $n_q > 0$;

3) если $n_p = 0$, то $q - 1$ делит $p - 1$ и достигается верхняя оценка, то есть $h_c(G) = p + \dfrac{p-1}{q-1}$.

Доказательство. Пусть теперь $p > q \geq 2$, в этом случае ациклическая группа G порядка pq состоит из нейтрального элемента и некоторого числа элементов порядков p и q. Обозначим через n_p и n_q соответственно числа элементов порядков p и q в c-базисе группы G. По лемме 1 $h_c(G) = n_p + n_q$ и

$$pq = 1 + n_p(p - 1) + n_q(q - 1). \tag{7}$$

1) Так как $p > q$, то из равенства (7) получаем

$$(n_p + n_q)(q - 1) \leq pq - 1 \leq (n_p + n_q)(p - 1).$$

Отсюда с учетом леммы 1 следует:

$$q + \frac{q-1}{p-1} \leq h_c(G) \leq p + \frac{p-1}{q-1}.$$

Так как $h_c(G)$ – целое число, то нижнюю оценку можно уточнить, заменив на 1 положительную дробь $\dfrac{q-1}{p-1}$, которая меньше единицы.

2) Если $n_q = 0$, то из (7) получаем, что $h_c(G) = \dfrac{pq-1}{p-1} = q + \dfrac{q-1}{p-1}$. Имеем противоречие, так как $h_c(G)$ – целое число, в то время как $\dfrac{q-1}{p-1}$ не целое. Значит, $n_q > 0$.

3) Если $n_p = 0$, то из (7) получаем, что

$$h_c(G) = \frac{pq-1}{q-1} = p + \frac{p-1}{q-1}. \tag{8}$$

Отсюда, если $q - 1$ не делит $p - 1$, то имеем противоречие, так как $h_c(G)$ – целое число, в то время как $\dfrac{p-1}{q-1}$ не целое. Значит, если $n_p = 0$, то $q - 1$ делит $p - 1$ и выполнено (8), то есть достигается верхняя оценка для $h_c(G)$.

4. О построении класса конечных перестановочных автоматов, группы которых имеют *c*-ширину, превышающую заданную величину

Пусть $A = (V_1, V_n, h)$ – конечный автомат Мили без выходов со входным алфавитом $V_1 = \{0, 1\}$, с множеством состояний $V_n = \{(v_1, \ldots, v_n)\}$ (двоичные n-мерные векторы) и с функцией переходов $h: V_n \to V_n$, определяемой при входном двоичном символе x формулой

$$h(x, v_1, \ldots, v_n) = x \cdot g_1(v_1, \ldots, v_n) \oplus \bar{x} \cdot g_2(v_1, \ldots, v_n),$$

где g_1, g_2 – подстановки множества V_n. Группа G_A автомата A есть $\langle g_1, g_2 \rangle$.

Укажем условия, при которых группа G_A автомата A имеет c-ширину не меньше 2^n.

Утверждение 5. Пусть число $2^n - 1$ – простое и g_1, g_2 – линейные подстановки максимального периода, где $\langle g_1 \rangle \neq \langle g_2 \rangle$. Тогда $h_c(G_A) \geq 2^n$.

Доказательство. Каждый из элементов g_1 и g_2, порождающих группу G_A, порождает максимальную циклическую подгруппу группы G_A. Действительно, по условию G_A – группа линейных преобразований множества V_n, поэтому всякая ее циклическая подгруппа порядка $2^n - 1$, в том числе группа $\langle g_1 \rangle$ и группа $\langle g_2 \rangle$, является максимальной в группе G_A. Следовательно, с учетом неравенства $\langle g_1 \rangle \neq \langle g_2 \rangle$ получаем по теореме 1, что в группе G_A имеется c-базис, включающий и элемент g_1, и элемент g_2. Отсюда, учитывая простоту числа $\operatorname{ord} g_2 = 2^n - 1$, по следствию 2 теоремы 1 получаем оценку: $h_c(G_A) > \operatorname{ord} g_2 = 2^n - 1$. Следовательно, $h_c(G_A) \geq 2^n$.

Выводы. Для конечных ациклических групп получены следующие результаты:

1) показано, что c-ширина любой группы не меньше 3, если порядок группы не делится на 4, то ее c-ширина больше 3;

2) описано строение групп, c-ширина которых равна 3;

3) показано, что c-ширина любой группы G не меньше $\dfrac{|G|-1}{\omega(G)-1}$, где $\omega(G)$ – наибольший из порядков элементов группы G;

4) c-ширина группы оценена снизу через числовые характеристики, определяемые порядками некоторых элементов c-базиса группы;

5) оценена c-ширина групп порядка pq, где p и q – простые числа, $p \leq q$, через числа p и q:

$$q + 1 \leq h_c(G) \leq p + \frac{p-1}{q-1},$$

в частности, при $p = q$ c-ширина группы в точности равна $p + 1$;

6) указан способ построения групп линейных подстановок множества V_n, у которых c-ширина не меньше 2^n.

ЛИТЕРАТУРА

1. *Биркгоф Г.* Теория решёток. М.: Наука, 1984. 567 с.
2. *Глухов М.М.*, *Елизаров В.П.*, *Нечаев А.А.* Алгебра. Т. 1. М.: Гелиос-АРВ, 2003. 336 с.
3. *Гретцер Г.* Общая теория решёток. М.: Мир, 1982. 454 с.
4. *Фомичёв В.М.* Исследование признаков в конечных группах и в группах подстановок // Математические вопросы кибернетики. Вып. 14: Сб. статей / Под ред. О.Б. Лупанова. М.: Физматлит, 2005. С. 161 – 260.
5. *Фомичёв В.М.* О вычислительной сложности определения характеристик наследственного подмножества группы с заданным признаком // Безопасность информационных технологий. М.: МИФИ, 2006. №2. С. 82 – 85.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

DOI 10.17223/20710410/2/7

УДК 519.7

О ПРИМЕНЕНИЯХ КВАЗИГРУПП В КРИПТОГРАФИИ

М.М. Глухов

Академия криптографии РФ, г. Москва

E-mail: glukhovmm@rambler.ru

Приводится краткий обзор опубликованных результатов по рассматриваемому вопросу, в том числе по применению квазигрупп для построения схем аутентификации, шифрования и однонаправленных функций.

***Ключевые слова*:** *квазигруппа*, *латинский квадрат*, *код аутентификации*, *шифр*, *однонаправленная функция.*

1. Схемы аутентификации

Первые известные нам работы с явным указанием на применение квазигрупп, или латинских квадратов, в криптографии относятся к построению кодов аутентификации, или А-кодов (см. [1, 2 – 7, 13, 14, 24 – 25, 27]). Во всех этих работах обсуждается и модифицируется схема, предложенная в работе [3]. Опишем основную конструкцию этой схемы по работе [7]. В ней в качестве сообщений рассматриваются слова длины, кратной m, в q-ичном алфавите Q, а в качестве подписи к сообщению s длины mt – слово $a_1 \ldots a_m$ длины m в том же алфавите, образованное следующим образом. Слово s разбивается по некоторому правилу на t слов $s^{(1)}, \ldots, s^{(t)}$ длины m, и по слову $s^{(i)}$ вырабатывается буква a_i с помощью квазигруппы $(Q, *)$, которая является ключом. А именно, если $s^{(i)} = s_1 \ldots s_m$, то буква a_i находится по формуле

$$a_i = (\ldots((s_1 * s_2) * s_3) * \ldots) * s_m.$$

Пользуясь лишь определением квазигруппы, легко показать, что произведение $(\ldots((s_1 * s_2) * s_3) * \ldots) * s_m$ принимает в качестве значений каждый элемент из Q одно и то же число раз. Отсюда следует, что при равномерном распределении вероятностей на множестве сообщений построенный А-код имеет минимальную вероятность успешной имитации, равную $1/q^m$. Особо рассматривается практически наиболее интересный, двоичный, случай, когда $q = 2^k$. Приводится естественная модификация схемы в случае, когда длина сообщений не кратна m. Исследуется также вариант, в котором алфавит подписи имеет большую мощность по сравнению с алфавитом сообщений.

В [4] анализируется возможность восстановления сообщения или ключевой квазигруппы в указанной схеме по ряду сообщений с определенными совпадениями букв в сообщениях и подписях. Рассматриваются два метода. В первом из них слова $s^{(1)}, \ldots, s^{(t)}$ являются отрезками слова-сообщения s, во втором – предполагается, что сообщение имеет длину q^2, t полагается равным q, а слова $s^{(1)}, \ldots, s^{(t)}$ образуются с использованием сообщения s и квазигруппы $(Q, *)$. Указываются условия, при которых в первом случае возможно восстановление открытого текста и даже ключевой квазигруппы. Схема второго типа является более стойкой к рассмотренной атаке.

В работе [4] рассматривается также вопрос об эквивалентности ключей, который сводится к рассмотрению определенных автотопий квазигруппы $(Q, *)$.

В [1] описанная выше схема аутентификации претерпевает дальнейшее усложнение с точки зрения образования слов $s^{(1)}, \ldots, s^{(t)}$. В ней рассматривается лишь случай, когда порядок q алфавита есть степень двойки, $t = q^2$, сообщение s имеет длину q^2 и представляется в виде квадратной матрицы M порядка q. По ключевой квазигруппе $(Q, *)$ строится система из $q/2$ попарно ортогональных квазигрупп, изотопных квазигруппе $(Q, *)$. С использованием матрицы M и таблиц Кэли имеющихся $(q/2) + 1$ квазигрупп и строится последовательность слов $s^{(1)}, \ldots, s^{(t)}$. Авторы высказывают гипотезу, что построенная ими схема в некотором смысле является оптимальной.

2. Шифры

Более общий подход к использованию квазигрупп для шифрования был предложен в работах [14, 19]. В [14] вводится так называемое квзигрупповое преобразование множества Q^+ всех конечных наборов букв конечного алфавита Q (Quasigroup String Processing). Оно определяется следующим образом. Сначала по за-

данной квазигруппе $(Q, *)$ и любому ее элементу (лидеру) a определяется преобразование $E_a^{(1)}$ множества Q^+:

$$E_a^{(1)}(x_1, \ldots, x_k) = (y_1, \ldots, y_k), \quad (1)$$

где $y_1 = a * x_1$, $y_{i+1} = y_i * x_{i+1}$, $i = 1, \ldots, k-1$. Затем берется композиция n таких преобразований, соответствующих квазигруппам $(Q, *_i)$ и выбранным лидерам a_i, $i = 1, \ldots, n$. Итоговое преобразование обозначается в виде $E^{(n)}_{a_n,\ldots,a_1}$. Преобразование $E_a^{(1)}$ обратимо, и обратное к нему преобразование $D_a^{(1)}$ определяется следующим образом:

$$D_a^{(1)}(x_1, \ldots, x_k) = (y_1, \ldots, y_k),$$

где $y_1 = a \setminus x_1$, $y_{i+1} = x_i \setminus x_{i+1}$, $i = 1, \ldots, k-1$. Отсюда легко находится обратное преобразование и для $E^{(n)}_{a_n,\ldots,a_1}$. Оно обозначается в виде $D^{(n)}_{a_1,\ldots,a_n}$. Предлагается использовать преобразование $E^{(n)}_{a_n,\ldots,a_1}$ для шифрования информации. При этом в качестве ключей предлагается использовать операции $*_i$.

В работах [14, 19] изучаются свойства соответственно преобразований $E^{(n)}_{a_n,\ldots,a_1}$ и $D^{(n)}_{a_n,\ldots,a_1}$ и делается вывод о том, что указанные преобразования обладают некоторыми нужными для криптографии качествами. В частности, при любых фиксированных $(Q, *_i)$, a_i, c_i, $i = 1, \ldots, n$, уравнение

$$E^{(n)}_{a_n,\ldots,a_1}(x_1, \ldots, x_k) = (c_1, \ldots, c_k) \quad (2)$$

имеет единственное решение относительно неизвестных $x_1, \ldots, x_k$; по соотношению (2) при известных $a_1, \ldots, a_k, x_1, \ldots, x_k, c_1, \ldots, c_k$ для нахождения ключей $(*_i)$ необходимо произвести столько проб, сколько существует наборов из $n-1$ квазигрупповых операций; выходные последовательности обладают хорошими статистическими свойствами, а именно: при любом заданном вероятностном распределении букв во входных последовательностях предельные распределения s-грамм в выходных последовательностях длины k являются равномерными при $k \to \infty$ и $s \le n$.

В ряде работ квазигрупповые преобразования слов используются для построения поточных шифров, криптографическая стойкость которых основана на сложности решения таких задач, как факторизация целых чисел или дискретное логарифмирование в конечных полях. Рассмотрим один из таких шифров, предлагаемый и анализируемый в работе [16]. Он получается путем комбинирования шифра типа Эль-Гамаля и поточного квазигруппового шифра. Приведем его подробное описание.

Шифр типа Эль-Гамаля хорошо известен. В качестве поточного квазигруппового шифра берется шифр в алфавите $Q = \mathbf{Z}_p^*$ с функциями шифрования $E^{(n)}_{a_n,\ldots,a_1}$ и расшифрования $D^{(n)}_{a_1,\ldots,a_n}$ из [14, 19], которые обозначаются здесь в виде E_α и E_α^{-1} при $\alpha = (a_1, \ldots, a_n)$. Предполагается, что они определены для случая, когда все квазигруппы $(Q, *_i)$ совпадают с одной и той же квазигруппой $(Q, *)$.

В [16] указывается также и способ получения квазигрупповой операции $*$ на Q. Она определяется следующим образом. Для произвольного $K \in \{1, \ldots, p-2\}$ задается отображение f_K: $Q \to Q$ по формуле

$$f_K(j) = (1/(1 + (K + j) (\mathrm{mod}\ p-1)))(\mathrm{mod}\ p)$$

и полагается

$$i * j = i \cdot f_K(j) (\mathrm{mod}\ p).$$

Нетрудно проверить, что отображение f_K является подстановкой, а группоид $(Q, *)$ – квазигруппой с левой обратной операцией

$$i \setminus j = ((i \cdot j^{-1} (\mathrm{mod}\ p)) - 1 - K) (\mathrm{mod}\ p-1)$$

при условии, что вместо 0 берется $p-1$.

Предлагаемый алгоритм имеет две части. В нем установление связи, выработка и передача ключей осуществляется с помощью шифра Эль-Гамаля, а шифрование исходного открытого сообщения и его расшифрование – с помощью квазигруппового шифра. В итоге получается существенный выигрыш в скорости по сравнению с известными асимметричными шифрами.

Проблема стойкости для первой части шифра решается точно так же, как для шифра Эль-Гамаля. Поэтому автор анализирует стойкость лишь второй части шифра, а именно: рассматривается задача нахождения числа K, определяющего квазигруппу $(Q, *)$, и лидеров $a_i \in Q$, $i = 1, \ldots, k$, по открытому тексту и соответствующему ему шифртексту. Показано, что эта задача сводится к решению системы полиномиальных уравнений над полем $\mathbf{Z}_p$ при большом простом числе p. Вид такой системы зависит от числа k используемых лидеров. При $k = 1$ и $k = 2$ получаются соответственно системы уравнений 2-й и 3-й степеней, и они могут быть эффективно решены. При $k > 2$ получаются системы уравнений более высоких степеней. В общем случае эта задача является NP-трудной. В рассматриваемом случае возникающие системы уравнений имеют определенную специфику, и задача их решения остается открытой.

Отметим еще, что в рассматриваемой работе для получения квазигрупповой операции, или, что то же самое, латинского квадрата, на $\mathbf{Z}_p^*$ указывается на способ, предложенный в [22].

В работе [22] предлагается использовать квазигруппы для построения преобразований типа «Все или ничего» («All-Or-Nothing Transformation», сокращенно AONT). Такие преобразования введены в 1997 г. Р. Райвестом с целью повышения стойкости блочных шифров с постоянной длиной блоков к методам, использующим свойства открытого текста.

Преобразование T последовательности сообщений (блоков) $m = (m_1, \ldots, m_s)$ в последовательность $m' = (m_1', \ldots, m_t')$, $t \geq s$, является *преобразованием типа AONT*, если T обратимо, T и T^{-1} эффективно вычислимы (т.е. вычислимы за полиномиальное время) и невозможно получение какой-либо информации о любом блоке m_i, если неизвестен хотя бы один блок m_j'.

В качестве AONT Райвест предложил так называемое *пакетное преобразование*, которое может быть описано следующим образом.

Дана входная последовательность сообщений $m = (m_1, \ldots, m_s)$, в которой m_i – отрезок из b бит, $i = 1, \ldots, s$.

1. Выбираем случайно ключ K для пакетного преобразования в блочном шифре.

2. Вычисляем последовательность $m' = (m_1', \ldots, m_{s+1}')$ по правилу

$$m_i' = m_i \oplus E_K(i),\ i = 1, \ldots, s;\ m_{s+1}' = K \oplus h_1 \oplus \ldots \oplus h_s,\ h_i = E_{K'}(m_i' \oplus i),\ i = 1, \ldots, s,$$

где K' – фиксированный открытый ключ, а E_K – функция зашифрования рассматриваемого блочного шифра при реальном ключе K.

Легко видеть, что пакетное преобразование обратимо, т.е. по m' однозначно находятся K и m. При этом если хотя бы один из блоков m_i' неизвестен, то невозможно вычислить K, а потому и любой блок m_i.

В отличие от сообщения m последовательность m' называют *псевдосообщением*. Предполагается, что далее псевдосообщение шифруется обычным образом. Таким образом, пакетное преобразование является предварительным преобразованием открытого сообщения. Оно не уменьшает криптографическую стойкость основного шифра и в то же время не позволяет найти исходное открытое сообщение при любом неполном дешифровании псевдосообщения. В последующие годы преобразования такого типа и их использование в криптографии рассматривались и обсуждались многими авторами (см., например, [2, 22, 26, 27]).

Следует иметь в виду, что методы шифрования с использованием AONT имеют и недостатки, а именно: AONT размножают ошибки и отрицательно влияют на скорость шифрования. По сути дела в схеме Райвеста пакетное преобразование требует не меньше времени, чем последующее шифрование псевдосообщения. Для борьбы с первым недостатком предлагается использовать корректирующие коды и хорошие каналы связи. Для уменьшения второго недостатка в работе [22] предлагается использовать новый способ построения псевдосообщения, основанный на комбинировании AONT и квазигруппового шифрования. С этой целью авторы предлагают следующий способ быстрого получения квазигрупп (латинских квадратов) порядка $n = p–1$, где p – простое число. В работе рассматривается случай $n = 256$, $p = 257$, и квазигруппа строится на множестве $Q = \{1, \ldots, 256\}$.

Сначала выбирается случайная перестановка $(a_{11}, \ldots, a_{1n})$ чисел из Q, которая принимается за первую строку латинского квадрата и таблицы Кэли нужной квазигруппы, затем строятся остальные строки по правилу

$$a_{ij} = i \cdot a_{1j} \pmod p,\ i = 2, \ldots, n;\ j = 1, \ldots, n.$$

Очевидно, что в итоге получится латинский квадрат и таблица Кэли квазигруппы $(Q, *)$.

Далее по сообщению $u = (u_1 \ldots u_k)$, где u_i – байты, являющиеся двоичными записями чисел из Q с условием, что 256 представляется байтом нулей, строится псевдосообщение $v = (v_1 \ldots v_k)$ следующим образом. Случайно выбирается произвольный элемент $a_1 \in Q$, называемый лидером, и полагается

$$v_1 = a_1 * u_1,\ v_{i+1} = v_i * u_{i+1},\ i = 2, \ldots, k.$$

После этого полученное псевдосообщение расширяется путем добавления перед ним лидера a_1 и первой строки квазигруппы. Далее расширенное псевдосообщение шифруется обычным образом.

Из построения расширенного псевдосообщения видно, что по нему однозначно восстанавливается исходное открытое сообщение. При этом используется левая обратная операция для операции $*$. Кроме того, видно также, что описанное предварительное преобразование открытого текста является преобразованием типа «Все или ничего».

3. Однонаправленные функции

Вернемся к вопросу об использовании квзигрупповых преобразований. В работе [29] предлагается использовать квазигрупповые преобразования для построения однонаправленных функций и подстановок. С этой целью автор выбирает квазигруппу $(Q, *)$ и определяет два преобразования

$$R_i : Q^N \to Q^N,\ i = 1, 2,$$

с использованием введенного ранее преобразования $e_a = E_a$ (см. (1)). А именно, для набора $A = (a_0, \ldots, a_{N-1})$ определяются:

$$R_1(A) = e_{a_{N-1}}(\ldots(e_{a_1}(e_{a_0}(A)))\ldots),$$

$$R_2(A) = e_{a_{N-1}}(...(e_{a_1}(e_{a_0}(R_1(A))))...) .$$

Автор доказывает, что если исходная квазигруппа не коммутативна и не ассоциативна, то для нахождения прообраза в преобразованиях R_1, R_2 при использовании лишь таблицы Кэли квазигруппы Q понадобится соответственно $O(q^{[N/2]})$ и $O(q^N)$ квазигрупповых операций. Исходя из этого, автор считает введенные функции серьезными кандидатами в однонаправленные функции. Преобразование R_2 используются также для построения других, более сложных, кандидатов в однонаправленные функции.

В работе [9] предлагается и исследуется вариант построения сжимающих хэш-функций итеративного типа с использованием в боксах так называемых мультиподстановок на множестве E^2, где

$$E = F_2^m = \{e_0, e_1, \ldots, e_t\}, t = 2^k - 1,$$

F_2 – поле из двух элементов. Предлагаемые функции, обозначаемые автором в виде $g_{k,s}$, являются равновероятными отображениями 2^t-й степени множества E в его 2^{t-1}-ю степень.

Функция $g_{k,s}$ определяется с использованием $2^{k-1}(s + 1)$ боксов $B_{i,j}$, $i = 0, 1, \ldots, 2^{k-1}-1$, $j = 0, 1, \ldots, s$, объединенных определенным образом в сеть, состоящую из $s+1$ ярусов по 2^{k-1} боксов в каждом ярусе. В ярус с номером j входят боксы $B_{i,j}$ со значениями

$$i = i_0 + 2i_1 + \ldots + 2^{k-1}i_{k-1} \text{ при } i_j = 0.$$

Боксы $B_{i,0}$ – входные, боксы $B_{i,s}$ – выходные. Каждый бокс осуществляет подстановку на множестве E^2.

Множество E разбивается на два подмножества H и M, где $H = \{e_i: i_0 = 0\}$ – множество хэш-входов, $M = \{e_i: i_0 = 1\}$ – множество входов-сообщений. На вход бокса $B_{i,j}$ подаются элементы $e_i \in H$ и $e_{i(j)} \in M$, отличающиеся лишь цифрами j-х разрядов. Предполагается, что на выходе блока $B_{i,j}$ получается также пара из $H \times M$, компоненты которой поступают соответственно на блоки $B_{i,j+1}$ и $B_{i(j),j+1}$. В итоге все входы-сообщения (а также и все хэш-входы) будут сниматься на каждом ярусе с разных блоков.

Далее, путем итерации преобразования $g_{k,s}$ авторы определяют искомую хэш-функцию $h_{k,s}$ следующим образом. Сообщение M в битах дополняется некоторым образом до числа бит, кратного $m2^{k-1}$, и представляется в виде $M = M_1M_2 \ldots M_n$, где M_i – набор длины $m2^{k-1}$. Теперь фиксируется начальное значение хэш-входа H_0 и вычисляются

$$H_i = g_{k,s}(H_{i-1}, M_i), i = 1, \ldots, n,$$

и в качестве значения хэш-функции $h_{k,s}$ (M) берется H_n.

В [8] исследуются свойства построенных таким образом хэш-функций в случае, когда в качестве боксов $B_{i,j}$ используются так называемые мультиподстановки множества E^2.

Под *мультиподстановкой* авторы понимают биективное отображение B: $E^2 \to E^2$, при котором $B(a, b) = (B_1(a, b), B_2(a, b))$ и функции B_1, B_2 являются подстановочными по каждому переменному. Это равносильно тому, что функции $B_1(a, b)$, $B_2(a, b)$ задают ортогональные квазигруппы на множестве E. В этом случае авторы указывают алгоритмы решения уравнения

$$h(H, M) = H'$$

относительно M при известных H, H' и нахождения коллизий. Находят верхние оценки сложности и выражают надежду, что эти оценки близки к нижним.

Для построения пар ортогональных квазигрупп авторы рекомендуют использовать ортоморфизмы групп. В частности, доказана теорема о том, что отображение L_c: $E^2 \to E^2$, при котором

$$L_c(a, b) = (a \oplus b, a \oplus bc \oplus R^n(b)),$$

где R – сдвиг вектора на 1 бит вправо, является мультиподстановкой тогда и только тогда, когда для любого $i \in \{1, \ldots, m\}$ среди векторов $R^{nt}(c)$, $t = 1, 2, \ldots, m$, найдутся векторы с различными i-ми координатами.

ЛИТЕРАТУРА

1. *Bakhtiari S*, *Safavi-Naini R.*, *Pieprzyk J.* A message Authentication Code based on Latin Squaresc // Proc. Australasian on Information Security and Privacy. 1997. P. 194 – 203.
2. *Canda V.*, *van Trung T.* A New Mode of Using All-Or-Nothing Transforms // Codes and Cryptography. 2001. P. 1 – 6.
3. *Carter S*, *Wegman M.N.* Universal Class of Hash Function // J. Computer and System Sciences. 1979. V. 18. No. 2. P. 143 – 154.
4. *Dawson E.*, *Donovan D*, *Offer A.* Quasigrops, isotopism and authentication schemes // The Australasian journal of combinatorics. 1996. V. 13. P. 75 – 88.
5. *Denes J.*, *Keedwell A.D.* Latin Squares. New Developments in the Theory and Applications. Amsterdam: Nord-Holland Publishing Co., 1981.
6. *Denes J.*, *Petroczki P.* A digital encrypting communication systems // Hungarian Patent. 1990. No. 201437A.
7. *Denes J.*, *Keedwell A.D.* A new Authentication Scheme based in Latin Squares // Discrete Mathematics. 1992. V. 106/107. P. 157 – 162.
8. *Vandenay S.* On the Need for Multipermutations: Cryptoanalysis of MD4 and SAFER // Proc. Fast Software Encryption. 1994. P. 286 – 297.

9. *Schnorr C.P.*, *Vandenay S.* Black box cryptoanalysis of hash networks based on multipermutations // Lecture Notes in Computer Science. 1995. V. 950. P. 47 – 57.
10. *Koscielny C. A* metod of constructing quasigroup-based stream-ciphers // Appl. Math. and Comp. Sci. 1996. V. 6. P. 109 – 121.
11. *Krawczyk H.* LFSR-based and Authentication // 1994, EUROCRYPT-94. P. 129 – 139.
12. *Krawczyk H.* New Hash Function for Message Authentication // 1995, EUROCRYPT-95. P. 301 – 310
13. *Markovski S.*, *Gligoroski D.*, *Andova S.* Using quasigroups for one-one secure encoding //Proc. VIII Conf. Logic and Comp. Sci. "LIRA 97", Novi Sad, 1997. P. 157 – 162.
14. *Markovski S.*, *Gligoroski D.*, *Bakeva V.* Quasigroup String Processing: Part 1 // Proc. of Maked. Academ. of Sci. and Arts for Math. And Tech. Sci. XX. 1 – 2. 1999. P. 13 – 28.
15. *Denes J. and Owens P.J.* Some new latin power sets not based on groups // J. Combin. Theory, ser A. 1999. V. 85. P. 69 – 82.
16. *Gligoroski D.* Stream cipher based on quasigroup string transformation in $\mathbf{Z}_p^*$ // Universitet "St. Cyril and Methodious", Faculty of Natural Sciences, Institute of Informatics, P.O. Box 162, Scopje, Republic of Macedonia. ArXiv:cs.CR/0403043 v2 22Apr 2004. gligoroski@yahoo.com
17. *Gligoroski D.*, *Markovski S.*, *Bakeva V*. Quasigroup and Hash Functions // Discr. Math. And Appl. Proc. of the 6th ICDMA, Bansko. 2001. P. 43 – 50.
18. *Gligoroski D.*, *Markovski S.*, *Bakeva V.* On infinite Class of strongly Collision Resistant Hash Functions "EDON-F" with Variable Length of Output // Proc. 1st International Conference on Mathematics and Informatics for Industry. Thessaloniki, Greece, 2003.
19. *Markovski S.*, *Kusacatov V.* Quasigroup String Processing: Part 2 // Proc. of Maked. Academ. of Sci. and Arts for Math. and Tech. Sci. XXI, 1 – 2. 2000. P. 15 – 32.
20. *Markovski S.*, *Gligoroski D.*, *Stojcevska B*. Secure two-way on-line communication by using quasigroup enciphering with almost public key // Novi Sad J. Mathematics. 2000. P. 30.
21. *Markovski S.* Quasigroup String Processing and Application in Cryptography // Invited talk. Proc. 1st International Conference on Mathematics and Informatics for Industry. Thessaloniki, Greece, 2003.
22. *Marnas S.I.*, *Angelis L.*, *Bleris G.L.* All-Or-Nothing Transform using quasigroups // Proc. 1st Balkan Conference in Informatics. 2003. P. 183 – 191.
23. *Rogaway P*. Bucket Hashing and its Application to Fast Message // 1995, CRYPTO-95. P. 30 – 42.
24. *Shoup V.* On Fast and Provably Sequre Message Authentication based on Universal Hashing // 1996, CRYPTO-96. P. 313 – 338.
25. *Stinson D.R.* Universal Hashing and Authentication Codes. Design // Codes and Cryptography. 1994. V. 4. P. 369 – 380.
26. *Stinson D.R.* Something about All-Or-Nothing (Transform). Design // Codes and Cryptography. 2001. P. 133 – 138.
27. *Taylor R.* Near optimal Unconditionally Sequre Authentication // 1994, EUROCRYPT-94. P. 245 – 255.
28. *Wegman M.N.*, *Carter S.* New Hash Functions and their Use in Authentication and Set Equality // J. Computer and System Sciences. 1981. V. 22. P. 265 – 279.
29. *Gligoroski D.* Candidate One-Way Functions and One-Way Permutations Based on Quasigroup String Transformations // 2005, gligoroski@yahoo.com

2008 Математические методы криптографии № 2(2)
DOI 10.17223/20710410/2/8

УДК 681.3

ПРЕДСТАВЛЕНИЕ КРИПТОСИСТЕМ МНОГООСНОВНОЙ АЛГЕБРАИЧЕСКОЙ СИСТЕМОЙ

Е.А. Иващенко, В.Г. Скобелев

Донецкий национальный университет,
Институт прикладной математики и механики НАН Украины, *г. Донецк*

E-mail: eugenia1000@rambler.ru, skbv@iamm.ac.donetsk.ua

На основе многоосновной алгебраической системы построена формальная модель криптосистемы. В рамках этой модели выделены основные типы криптосистем.

Ключевые слова: *модель криптосистемы*, *многоосновная алгебраическая система.*

Современные задачи криптографии обуславливают необходимость разработки математического аппарата для моделирования систем защиты информации, их сравнительной характеристики, анализа их вычислительной стойкости и имитостойкости. Системы защиты информации [1 – 3] характеризуются многообразием и сложностью процессов их взаимодействия с внешней средой, а также сложностью внешней среды (содержащей интеллектуальные компоненты). При этом математическая модель криптосистемы играет фундаментальную роль. Известны подходы к построению таких моделей с позиций теории систем [4] и современной алгебры [5].

Первый подход [1] базируется на системе вход-выходного типа

$$S = (M , C , K_1 , K_2 , E , D) ,$$

где M, C, K_1 и K_2 – множество соответственно открытых текстов, шифртекстов, ключей шифрования и ключей расшифровки, а $E : M \times K_1 \to C$ и $D : C \times K_2 \to M$ – алгоритмы шифрования и расшифровки. Достоинство этой модели – представление согласованности процессов шифрования и расшифровки биекцией $k : K_1 \to K_2$, возможность выделения блочных и поточных криптосистем, а также ряда портов, через которые осуществляются пассивные атаки криптоаналитика.

Второй подход [6] основан на алгебраической системе

$$S = (T , F, domain , range , F_e , F_c) ,$$

где T и F – множество имен соответственно типов и функций, $domain : F \to T^*$ и $range : F \to T$ – отображения, F_e ($F_e \subseteq F$) – множество легко вычислимых функций, а $F_c = \{ f \in F \mid domain (f) = \lambda \}$ (λ – пустое слово) – множество констант. Достоинство этой модели – возможность построения на множестве термов системы конгруэнций, предназначенной для формирования определяющих соотношений для конкретной криптосистемы, и выделения ряда портов, через которые осуществляются пассивные атаки криптоаналитика.

Однако обе модели имеют существенные недостатки. Во-первых, они могут представлять системы с предвосхищением и системы, содержащие невычислимые функции. Во-вторых, необходима их дополнительная проработка для выделения основных классов криптосистем [1] и представления основных типов атак криптоаналитика. В-третьих, они не дают возможность эффективно представлять параметрические криптосистемы [7, 8], нестационарные криптосистемы и криптосистемы с вариацией окна шифрования. Естественный путь устранения этих недостатков – это выбор в качестве базовой модели варианта системы алгоритмических алгебр [9]. Для этого необходимо построить соответствующую многоосновную алгебраическую систему. Решение этой задачи – основная цель настоящей работы.

Структура работы следующая: в п.1 построена и охарактеризована базовая многоосновная алгебраическая система, в п.2 в рамках этой модели выделены основные типы криптосистем. Заключение содержит ряд выводов.

1. Базовая алгебраическая система

Рассмотрим многоосновную алгебраическую систему

$$\mathsf{S} = (\mathsf{T} , \mathsf{F}),$$

где семейство T основных множеств и сигнатура F имеют вид

$$\mathsf{T} = \{ T_{ij} = \{ t_{ij}^{(r)} \mid r \in \mathbf{N} \} \mid i = 1, \ldots, 8 ; j = 1, 2 \},$$

$$\mathsf{F} = U \cup F \cup K \cup \Phi .$$

Предполагается, что множества T_{ij} попарно не пересекаются, причем T_{i2} ($i = 1, \ldots, 8$) так линейно упорядочены, что

$$t_{i2}^{(1)} < t_{i2}^{(2)} < \ldots < t_{i2}^{(n)} < \ldots$$

Множества $T_{11}, \ldots, T_{81}$ и $T_{12}, \ldots, T_{82}$ назовем множествами имен и множествами размеров соответственно открытых текстов, ключей шифрования, параметров шифрования, состояний шифрования, шифртекстов, ключей расшифровки, параметров расшифровки и состояний расшифровки.

Охарактеризуем теперь сигнатуру F, состоящую из имен легко вычислимых функций.

I. Множество U состоит из имен монотонно возрастающих функций и имеет вид

$$U = \{ u_i^{(1)} : \mathbf{N} \to \mathbf{N},\ u_i^{(2)} : \mathbf{N}^3 \to \mathbf{N},\ u_i^{(3)} : \mathbf{N}^3 \to \mathbf{N} \mid i = 1, \ldots, 8 \},$$

где для всех значений $i \in \{ 1, \ldots, 8 \}$ при любых фиксированных значениях $y, z \in \mathbf{N}$ ($z < y$) функция $v_i(x) = u_i^{(2)}(x, y, z)$ − кусочно-постоянная, а функция $w_i(x) = u_i^{(3)}(x, y, z)$ − периодическая на множестве $\{ 1, \ldots, u_i^{(1)}(y) \}$, и каждая из функций v_i, w_i отображает это множество на множество $\{ 1, \ldots, u_i^{(1)}(z) \}$.

Множество U предназначено для построения на каждом множестве $T_{i1}T_{i2}$ ($i = 1, \ldots, 8$) системы определяющих соотношений вида

$$\begin{cases} t_{i1}^{(h)} t_{i2}^{(r)} = t_{i1}^{(h - u_i^{(1)}(r))} t_{i2}^{(r)}, & \text{если } h > u_i^{(1)}(r), \\ t_{i1}^{(h)} t_{i2}^{(r)} = t_{i1}^{(h_1)} t_{i2}^{(r_1)} t_{i1}^{(h_2)} t_{i2}^{(r_2)}, \end{cases} \tag{1}$$

где

$$r = r_1 + r_2 \ (r_1, r_2) \in \mathbf{N},$$

$$h_1 = u_i^{(2)}(h, r, r_1),$$

$$h_2 = u_i^{(3)}(h, r, r_2).$$

Первое из соотношений (1) осуществляет переход от множества $T_{i1}T_{i2}$ ($i = 1, \ldots, 8$) к множеству

$$T_{i,12} = \{ t_{i1}^{(h)} t_{i2}^{(r)} \mid r \in \mathbf{N},\ h \in \{ 1, \ldots, u_i^{(1)}(r) \} \} \subset T_{i1}T_{i2}.$$

Положим

$$T_{i,12}(n) = \{ t_{i1}^{(h)} t_{i2}^{(n)} \in T_{i,12} \mid h \in \{ 1, \ldots, u_i^{(1)}(n) \} \} \ (n \in \mathbf{N}).$$

Тогда

$$T_{i,12} = \bigcup_{n=1}^{\infty} T_{i,12}(n),$$

где $T_{i,12}(n)$ ($n \in \mathbf{N}$) – попарно непересекающиеся конечные множества. Значение второго из соотношений (1) состоит в следующем. Систему соотношений (1) назовем полугрупповой, если каждый элемент $t_{i1}^{(h)} t_{i2}^{(r)} \in T_{i,12}$ ($i = 1, \ldots, 8$) единственным образом представляется в виде

$$t_{i1}^{(h)} t_{i2}^{(r)} = t_{i1}^{(h_1)} t_{i2}^{(1)} \ldots t_{i1}^{(h_r)} t_{i2}^{(1)}.$$

Пусть $X = \{ x_1, \ldots, x_m \}$ ($m \in \mathbf{N}$), $t_{i2}^{(r)} = r$ ($r \in \mathbf{N}$) и $u_i^{(1)}(r) = m^r$. Определим биекцию $\varphi : T_{i,12} \to X^+$ равенствами

$$\varphi(t_{i1}^{(h)} t_{i2}^{(r)}) = x_{j_1} \ldots x_{j_r} \quad (h = 1, \ldots, u_i^{(1)}(r)) \ (r \in \mathbf{N}),$$

где

$$h = j_r + (j_{r-1} - 1)\, m + (j_{r-2} - 1)\, m^2 + \ldots + (j_1 - 1)\, m^{r-1}.$$

В этом случае соотношения (1) согласуются с лексикографическим порядком на каждом множестве $T_{i,12}(r)$ ($r \in \mathbf{N}$). Отсюда вытекает

Теорема 1. Полугрупповая система определяющих соотношений (1) непротиворечива.

Проиллюстрируем достоинства полугрупповой системы определяющих соотношений (1) на следующем простом примере.

Пример 1. 1. Пусть

$$X = \{000, 001, 010, 011, 100, 101, 110, 111\}.$$

Первое соотношение (1) дает возможность представить сообщение $t_{11}^{(22)} t_{12}^{(1)}$ в виде

$$t_{11}^{(20)} t_{12}^{(1)} = t_{11}^{(12)} t_{12}^{(1)} = t_{11}^{(4)} t_{12}^{(1)} = 011.$$

2. Пусть

$$X = \mathbf{Z}_4 = \{000, 001, 010, 011\}.$$

Второе соотношение (1) дает возможность представить сообщение $t_{11}^{(21)}t_{12}^{(2)}$ в виде

$$t_{11}^{(21)}t_{12}^{(2)} = t_{11}^{(5)}t_{12}^{(2)} = t_{11}^{(2)}t_{12}^{(1)}t_{11}^{(4)}t_{12}^{(1)} = 001000 .$$

Всюду в дальнейшем считаем, что система определяющих соотношений (1) – полугрупповая и истинны равенства

$$u_i^{(1)} = u_{i+4}^{(1)} \quad (i = 1, \ldots, 4) .$$

II. Множество F имеет вид

$$F = F_1 \cup F_2 ,$$

где F_1 и F_2 – равномощные непересекающиеся множества имен функций, удовлетворяющие следующим трем условиям.

Условие 1. Для каждого $f_j \in F_j$ $(j = 1, 2)$ существуют такие числа $n_{f_j}^{(1)}$, $n_{f_j}^{(2)} \in \mathbf{N}$, что

$$\mathrm{Dom}\, f_j = (\bigcup_{n=1}^{\infty} (T_{4j-3,12}(n) \times T_{4j-2,12}(n))) \times T_{f_j} ,$$

$$\mathrm{Val}\, f_j = \bigcup_{n=1}^{\infty} T_{9-4j,12}(n),$$

где $$\varnothing \neq T_{f_j} \subseteq T_{4j-1,12}(n_{f_j}^{(1)}) \times T_{4j,12}(n_{f_j}^{(2)}) .$$

Условие 2. Для всех $f_j \in F_j$ $(j = 1, 2)$ и $(t_{4j-3}, t_{4j-2}) \in T_{4j-3,12}(n) \times T_{4j-2,12}(n)$ $(n \in \mathbf{N})$,

$$f_j(t_{4j-3}, t_{4j-2}, t_{4j-1}, t_{4j}) \in T_{9-4j,12}(n)$$

при всех $(t_{4j-1}, t_{4j}) \in T_{f_j}$.

Условие 3. Для всех $f_j \in F_j$ $(j = 1, 2)$ функция

$$g_{f_j, t_{4j-2}, t_{4j-1}, t_{4j}}(t_{4j-3}) = f_j(t_{4j-3}, t_{4j-2}, t_{4j-1}, t_{4j,12})$$

является биекцией множества $T_{4j-3,12}(n)$ $(n \in \mathbf{N})$ на множество $T_{9-4j,12}(n)$ при всех фиксированных значениях $(t_{4j-2}, t_{4j-1}, t_{4j}) \in T_{4j-2,12}(n) \times T_{f_j}$.

Назовем F_1 множеством схем шифрования, а F_2 – множеством схем расшифровки.

III. Множество

$$K = \{\kappa_1, \kappa_2\}$$

состоит из имен таких биекций

$$\kappa_j : F_j \to F_{3-j} \quad (j = 1, 2) ,$$

что для всех $f_j \in F_j$ $(j = 1, 2)$ истинны равенства

$$|pr_1 T_{f_j}| = |pr_1 T_{\kappa_j(f_j)}| ,$$

$$|pr_2 T_{f_j}| = |pr_2 T_{\kappa_j(f_j)}| ,$$

$$|T_{f_j}| = |T_{\kappa_j(f_j)}| ,$$

IV. Множество

$$\Phi = \bigcup_{j=1}^{2} \bigcup_{f_j \in F_j} \{\alpha_{f_j,n}, \beta_{f_j}, \gamma_{f_j} \mid n \in \mathbf{N}\}$$

состоит из имен таких биекций

$$\alpha_{f_j,n} : T_{4j-2,12}(n) \to T_{(4j+2)(\mathrm{mod}\ 8),12}(n) ,$$

$$\beta_{f_j} : pr_1 T_{f_j} \to pr_1 T_{\kappa_j(f_j)} ,$$

$$\gamma_{f_j} : pr_2 T_{f_j} \to pr_2 T_{\kappa_j(f_j)} ,$$

что для всех $f_j \in F_j$ $(j = 1, 2)$ и $n \in \mathbf{N}$ равенства

$$\kappa_j(f_j)(f_j(t_{4j-3}, t_{4j-2}, t_{4j-1}, t_{4j}), \alpha_{f_j,n}(t_{4j-2}), \beta_{f_j}(t_{4j-1}), \gamma_{f_j}(t_{4j})) = t_{4j-3} \qquad (2)$$

истинны при всех $(t_{4j-3}, t_{4j-2}, t_{4j-1}, t_{4j}) \in T_{4j-3,12}(n) \times T_{4j-2,12}(n) \times T_{f_j}$.

Отметим, что равенства (2) обеспечивают взаимно-однозначное соответствие между результатами процессов шифрования и расшифровки.

Охарактеризуем теперь построенную алгебраическую систему S.

Для всех $f_j \in F_j$ $(j = 1, 2)$ и $n \in \mathbf{N}$ определим отношения эквивалентности

$$\varepsilon_1(f_j, n) \subseteq T_{4j-2,12}(n) \times T_{4j-2,12}(n),$$

$$\varepsilon_2(f_j) \subseteq pr_1 T_{f_j} \times pr_1 T_{f_j},$$

$$\varepsilon_3(f_j) \subseteq pr_2 T_{f_j} \times pr_2 T_{f_j}$$

следующим образом:

$$(t_{4j-2}, t'_{4j-2}) \in \varepsilon_1(f_j, n) \Leftrightarrow$$

$$\Leftrightarrow (\forall (t_{4j-3}, t_{4j-1}, t_{4j}) \in T_{4j-3,12}(n) \times T_{f_j})(f_j(t_{4j-3}, t_{4j-2}, t_{4j-1}, t_{4j}) =$$

$$= f_j(t_{4j-3}, t'_{4j-2}, t_{4j-1}, t_{4j})); \tag{3}$$

$$(t_{4j-1}, t'_{4j-1}) \in \varepsilon_2(f_j) \Leftrightarrow$$

$$\Leftrightarrow (\forall n \in \mathbf{N})(\forall (t_{4j-3}, t_{4j-2}, t_{4j}) \in T_{4j-3,12}(n) \times T_{4j-3,12}(n) \times \mathrm{pr}_2 T_{f_j})(f_j(t_{4j-3}, t_{4j-2}, t_{4j-1}, t_{4j}) =$$

$$= f_j(t_{4j-3}, t_{4j-2}, t'_{4j-1}, t_{4j})); \tag{4}$$

$$(t_{4j}, t'_{4j}) \in \varepsilon_2(f_j) \Leftrightarrow$$

$$\Leftrightarrow (\forall n \in \mathbf{N})(\forall (t_{4j-3}, t_{4j-2}, t_{4j-1}) \in T_{4j-3,12}(n) \times T_{4j-2,12}(n) \times \mathrm{pr}_1 T_{f_j})(f_j(t_{4j-3}, t_{4j-2}, t_{4j-1}, t_{4j}) =$$

$$= f_j(t_{4j-3}, t_{4j-2}, t_{4j-1}, t'_{4j})). \tag{5}$$

Из (2) – (5) вытекает

Теорема 2. Для всех $f_j \in F_j$ $(j = 1, 2)$:

1) если $(t_{4j-2}, t'_{4j-2}) \in \varepsilon_1(f_j, n)$ $(n \in \mathbf{N})$, то $(\alpha_{f_j,n}(t_{4j-2}), \alpha_{f_j,n}(t'_{4j-2})) \in \varepsilon_1(\kappa_j(f_j), n)$;

2) если $(t_{4j-1}, t'_{4j-1}) \in \varepsilon_2(f_j)$, то $(\beta_{f_j}(t_{4j-1}), \beta_{f_j}(t'_{4j-1})) \in \varepsilon_2(\kappa_j(f_j), n)$;

3) если $(t_{4j}, t'_{4j}) \in \varepsilon_3(f_j)$, то $(\gamma_{f_j}(t_{4j-1}), \gamma_{f_j}(t'_{4j-1})) \in \varepsilon_3(\kappa_j(f_j), n)$.

Отметим, что рассмотренные выше понятия дают возможность выделить следующие классы систем S:

1) класс слабо F-минимальных систем S, характеризующийся тем, что для любых двух элементов $f_j, f'_j \in F_j$ $(j = 1, 2)$ при любом $(t_{4j-1}, t_{4j}) \in T_{f_j} \cap T_{f'_j}$ существует такое число $n \in \mathbf{N}$ и такой элемент $t_{4j-2} \in T_{4j-2,12}(n)$, что

$$g_{f_j, t_{4j-2}, t_{4j-1}, t_{4j}} \neq g_{f'_j, t_{4j-2}, t_{4j-1}, t_{4j}};$$

2) класс сильно F-минимальных систем S, характеризующийся тем, что для любых двух элементов $f_j, f'_j \in F_j$ $(j = 1, 2)$

$$g_{f_j, t_{4j-2}, t_{4j-1}, t_{4j}} \neq g_{f'_j, t_{4j-2}, t_{4j-1}, t_{4j}}$$

при любом $(t_{4j-1}, t_{4j}) \in T_{f_j} \cap T_{f'_j}$ для всех $t_{4j-2} \in T_{4j-2,12}(n)$ $(n \in \mathbf{N})$;

3) класс K-минимальных систем S, характеризующихся тем, что

$$\kappa_1^{-1} = \kappa_2;$$

4) класс K-минимальных систем S, характеризующийся тем, что для всех $f_j \in F_j$ $(j = 1, 2)$ каждое отношение $\varepsilon_1(f_j, n)$ $(n \in \mathbf{N})$ – отношение равенства на множестве $T_{4j-2,12}(n)$.

2. Классы криптосистем

Для каждого $f_j \in F_j$ $(j = 1, 2)$ и $(t_{4j-1}, t_{4j}) \in T_{f_j}$ определим отображение

$$A_{f_j, t_{4j-1}, t_{4j}} : \bigcup_{n=1}^{\infty} (T_{4j-3,12}(n) \times T_{4j-2,12}(n)) \to \bigcup_{n=1}^{\infty} T_{9-4j,12}(n) \quad (j = 1, 2)$$

равенством $$A_{f_j, t_{4j-1}, t_{4j}}(t_{4j-3}, t_{4j-2}) = f_j(t_{4j-3}, t_{4j-2}, t_{4j-1}, t_{4j}).$$

Отображение A_{f_1, t_3, t_4} $(f_1 \in F_1, (t_3, t_4) \in T_{f_1})$ назовем алгоритмом (f_1, t_3, t_4)-шифрования, а отображение A_{f_2, t_7, t_8} $(f_2 \in F_2, (t_7, t_8) \in T_{f_2})$ – алгоритмом (f_2, t_7, t_8)-расшифровки.

Отметим, что такое определение алгоритмов шифрования и расшифровки дает возможность в терминах многоосновной алгебраической системы S выделить следующие три уровня понятия «ключ»:

1) элемент множества $T_{4j-2,12}$ интерпретируется как сеансовый или, иными словами, кратковременный ключ;

2) элемент множества $pr_2\, T_{f_j}$ интерпретируется как ключ средней длительности, т.е. как ключ, применяемый для определенного числа сеансов;

3) элемент множества $pr_1\, T_{f_j}$ интерпретируется как долговременный ключ.

Определим стационарную (f_1, t_3, t_4)-криптосистему $(f_1 \in F_1, (t_3, t_4) \in T_{f_1}))$ как упорядоченную пару

$$C_{f_1,t_3,t_4} = (A_{f_1,t_3,t_4}, A_{\kappa_1(f_1),\beta_{f_1}(t_3),\gamma_{f_1}(t_4)}),$$

а стационарную (f_2, t_7, t_8)-криптосистему $(f_2 \in F_2, (t_7, t_8) \in T_{f_2})$ – как упорядоченную пару

$$C_{f_2,t_7,t_8} = (A_{\kappa_2(f_2),\beta_{f_2}(t_7),\gamma_{f_2}(t_8)}, A_{f_2,t_7,t_8}).$$

Отметим, что определение криптосистемы как упорядоченной пары дает возможность выделить (если такая необходимость возникает), что является приоритетным: процесс шифрования или процесс расшифровки.

Покажем, что в терминах многоосновной алгебраической системы S могут быть представлены основные классы криптосистем.

Стационарную (f_j, t_{4j-1}, t_{4j})-криптосистему $C_{f_j,t_{4j-1},t_{4j}}$ $(f_j \in F_j\ (j = 1, 2), (t_{4j-1}, t_{4j}) \in T_{f_j})$ назовем:

1) симметричной криптосистемой, если каждое $\alpha_{f_j,n}$ $(n \in \mathbf{N})$ – имя такой биекции, что $\alpha^{-1}_{f_j,n}$ – имя легковычислимой биекции;

2) асимметричной, если каждое $\alpha_{f_j,n}$ $(n \in \mathbf{N})$ – имя такой биекции, что в настоящее время не известен быстрый алгоритм вычисления биекции $\alpha^{-1}_{f_j,n}$, либо доказано, что такой алгоритм не существует;

3) криптосистемой с автоключом, если $t_{4j-2} \in T_{4j-3,12}$ – фиктивная переменная;

4) криптосистемой с внешним сеансовым ключом, если $t_{4j-2} \in T_{4j-3,12}$ – существенная переменная;

5) параметрической криптосистемой, если $t_{4j-1} \in pr_1\, T_{f_j}$ – существенный параметр для системы $C_{f_j,t_{4j-1},t_{4j}}$, т.е. существуют два таких элемента $t_{4j-1}, t'_{4j-1} \in pr_1\, T_{f_j}$, что

$$C_{f_j,t_{4j-1},t_{4j}} \neq C_{f_j,t'_{4j-1},t_{4j}};$$

6) блочной криптосистемой, если

$$A_{f_j,t_{4j-1},t_{4j}}(t_{4j-3}t'_{4j-3}, t_{4j-2}t'_{4j-2}) = A_{f_j,t_{4j-1},t_{4j}}(t_{4j-3}, t_{4j-2})\, A_{f_j,t_{4j-1},t_{4j}}(t'_{4j-3}, t'_{4j-2})$$

для всех $(t_{4j-3}, t_{4j-2}), (t'_{4j-3}, t'_{4j-2}) \in \bigcup_{n=1}^{\infty} (T_{4j-3,12}(n) \times T_{4j-2,12}(n))$;

7) схемой с предысторией, если существует хотя бы одна такая пара значений

$$(t_{4j-3}, t_{4j-2}), (t'_{4j-3}, t'_{4j-2}) \in \bigcup_{n=1}^{\infty} (T_{4j-3,12}(n) \times T_{4j-2,12}(n)),$$

что $A_{f_j,t_{4j-1},t_{4j}}(t_{4j-3}t'_{4j-3}, t_{4j-2}t'_{4j-2}) \neq A_{f_j,t_{4j-1},t_{4j}}(t_{4j-3}, t_{4j-2})\, A_{f_j,t_{4j-1},t_{4j}}(t'_{4j-3}, t'_{4j-2})$.

Выделим следующий подкласс класса схем с предысторией, являющийся предметом исследования классической криптографии. Стационарную (f_j, t_{4j-1}, t_{4j})-криптосистему с предысторией $C_{f_j,t_{4j-1},t_{4j}}$ $(f_j \in F_j\ (j = 1, 2), (t_{4j-1}, t_{4j}) \in T_{f_j})$ назовем стационарной поточной криптосистемой, если существует такая легко вычислимая функция

$$\delta_{f_j,t_{4j-1},t_{4j}} : T_{f_j} \times \bigcup_{n=1}^{\infty} (T_{4j-3,12}(n) \times T_{4j-2,12}(n)) \to T_{f_j}, \tag{6}$$

что для всех $(t_{4j-3}, t_{4j-2}), (t'_{4j-3}, t'_{4j-2}) \in \bigcup_{n=1}^{\infty} (T_{4j-3,12}(n) \times T_{4j-2,12}(n))$

$$A_{f_j,t_{4j-1},t_{4j}}(t_{4j-3}t'_{4j-3}, t_{4j-2}t'_{4j-2}) = A_{f_j,t_{4j-1},t_{4j}}(t_{4j-3}, t_{4j-2})\, A_{f_j,t_{4j-1},t'_{4j}}(t'_{4j-3}, t'_{4j-2}), \tag{7}$$

где $$t'_{4j} = \delta_{f_j,t_{4j-1},t_{4j}}(t_{4j}, (t_{4j-3}, t_{4j-2})). \tag{8}$$

Из (1) вытекает следующая

Теорема 3. Для любой стационарной поточной криптосистемы $C_{f_j,t_{4j-1},t_{4j}}$ ($f_j \in F_j$ ($j = 1, 2$), (t_{4j-1}, t_{4j}) $\in T_{f_j}$) отображение $A_{f_j,t_{4j-1},t_{4j}}$ – ограниченно-детерминированная функция.

Следующий пример показывает, что современные криптосистемы естественно укладываются в рамки построенной выше классификации криптосистем.

Пример 2. 1. Шифр Вернама представляет собой стационарную непараметрическую блочную криптосистему с внешним сеансовым ключом.

2. Шифры Виженера представляют собой стационарные параметрические криптосистемы с предысторией и автоключом, причем роль параметра играет пароль.

3. Шифры DES, AES и ГОСТ 28147-89 представляют собой стационарные блочные параметрические криптосистемы с внешним сеансовым ключом. Для DES и ГОСТ 28147-89 параметром является набор S-блоков, а для AES – набор коэффициентов многочленов.

4. Шифр RSA представляет собой стационарную блочную параметрическую криптосистему с автоключом, причем роль параметра играет набор натуральных чисел.

5. Шифр RC4 представляет собой стационарную поточную параметрическую криптосистему с внешним сеансовым ключом, причем параметр – перестановка чисел 0, 1, …, 255.

6. Нелинейный БПИ-автомат над конечным кольцом [7, 8] представляет собой стационарную поточную параметрическую криптосистему с автоключом, причем параметр – это набор коэффициентов многочленов.

7. Любой квантовый шифр представляет собой криптосистему с предысторией.

Заключение

В работе построена многоосновная алгебраическая система S, предназначенная для исследования современных криптосистем с единых позиций. Дальнейшее, более глубокое исследование абстрактных свойств алгебраической системы S − одно из возможных направлений дальнейших исследований. Второе направление исследований связано с детальной проработкой введенного в работе понятия стационарная (f_j, t_{4j-1}, t_{4j})-криптосистема ($f_j \in F_j$ ($j = 1, 2$), (t_{4j-1}, t_{4j}) $\in T_{f_j}$). Третье направление исследований связано с построением и анализом в рамках алгебраической системы S формальной модели нестационарной криптосистемы. Четвертое направление исследований связано с разработкой в рамках алгебраической системы S формальных моделей пассивных и активных атак криптоаналитика, достаточных для теоретического анализа с единых позиций их эффективности и сложности. Пятое направление исследований связано с разработкой структуры данных, достаточной для эффективного компьютерного моделирования атак криптоаналитика на криптосистемы, определяемые в терминах алгебраической системы S.

ЛИТЕРАТУРА

1. *Алферов А.П.* и др. Основы криптографии. М.: Гелиос АРВ, 2002. 480 с.
2. *Молдовян А.А. и др*. Криптография. Скоростные шифры. СПб.: БХВ-Петербург, 2002. 496 с.
3. *Диффи У*., *Хеллмен М.Е*. Защищенность и имитостойкость: Введение в криптографию // ТИИЭР. 1979. Т. 67. № 3. С. 71 – 109.
4. *Месарович М.*, *Такахара Я.* Общая теория систем: математические основы. М.: Мир, 1978. 311 с.
5. *Мальцев А.И*. Алгебраические системы. М.: Наука, 1970. 329 с.
6. *Lynch N*. I/O automaton models and proofs for shared-key communication systems // Proceedings of the 12th IEEE Computer Security Foundations Workshop (CSFW'99), Mordana, Italy, June, 28 – 30, 1999. – 16 p.
7. *Скобелев В.Г*. Нелинейные автоматы над конечным кольцом // Кибернетика и системный анализ. 2006. № 6. С. 29 – 42.
8. *Скобелев В.Г*. О некоторых свойствах нелинейных БПИ-автоматов над кольцом $\mathbf{Z}_p^k$ // Прикладная радиоэлектроника. 2007. Т. 6. № 2. С. 288 – 299.
9. *Глушков В.М*., *Цейтлин Г.Е*., *Ющенко Е.Л*. Алгебра, языки, программирование. Киев: Наукова думка, 1978. 320 с.

2008 Математические методы криптографии № 2(2)

DOI 10.17223/20710410/2/9

УДК 62-50:519.7

ОБРАТИМЫЕ ДИНАМИЧЕСКИЕ СИСТЕМЫ С ПЕРЕМЕННОЙ РАЗМЕРНОСТЬЮ ФАЗОВОГО ПРОСТРАНСТВА В ЗАДАЧАХ КРИПТОГРАФИЧЕСКОГО ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ

А.М. Ковалев, В.А. Козловский, В.Ф. Щербак

Институт прикладной математики и механики НАН Украины, г. Донецк

E-mail: {kovalev, kozlovskii, shvf}@iamm.ac.donetsk.ua

Рассматривается способ преобразования информации, основанный на применении теории обратимых систем управления. Предложен метод изменения размерности пространства состояний, усложняющий поведение системы без изменения структуры уравнений. Тем самым вводятся структурные ключи, которые также могут быть секретными.

Ключевые слова: *обратные системы управления*, *защита информации*, *автомат*, *инвариантное множество.*

В последнее время различные аспекты теории нелинейных динамических систем со сложным (хаотическим) поведением траекторий находят применение в области обработки и защиты информации. В данной работе реализован подход, связанный с использованием методов теории обратимых систем управления в коммуникационных технологиях [1, 2]. В работе рассматривается метод преобразования и передачи информации, использующий дуальное нелинейное преобразование информации по следующей схеме: оцифрованное сообщение подается как внешнее воздействие на вход динамической системы, информация о ее траекториях, неявно зависящая от входа, в виде сигнала направляется в коммуникационные сети. В случае наличия у передающей системы свойства обратимости может быть синтезирована обратная система, играющая роль дешифратора (приемник). В работе показано, что при использовании обратных систем может возникнуть эффект, названный динамической деградацией. Он связан с возможностью попадания траекторий на инвариантные множества в фазовом пространстве, лежащие на многообразиях меньшей размерности. В общем случае это оказывает негативное влияние на сложность поведения и, таким образом, уменьшает стойкость соответствующих алгоритмов шифрования к определенным видам атак. Поэтому представляет интерес задача нахождения способов компенсации таких вырождений динамики. Для дискретных систем в рамках этой схемы предложен метод управления размерностью пространства состояний и/или входов.

1. Обратимые динамические системы

Предположим, что передатчик является дискретной динамической системой, правые части которой зависят от вектора функции $u(.)$ – оцифрованного информационного сообщения:

$$x(t+1) = f(x(t), u(t)), x(0) = x_0; \quad (1)$$

$$y(t) = h(x(t), u(t)), \quad (2)$$

где $x(t) \in \mathrm{R}^n$, $u(t) \in \mathrm{R}^m$, $y(t) \in \mathrm{R}^p$ определяют векторы состояния системы, ее вход и выход соответственно. По каналам связи передается выходной сигнал – функция $y(t)$, зависящая от состояния системы, ее параметров и сообщения $u(t)$. Для построения уравнений, описывающих динамику приемного устройства, рассмотрим задачу восстановления значений входного воздействия по значениям функции выхода. В теории управления непрерывными динамическими системами одним из способов ее решения является построение системы, обратной к исходной [2, 3]. В отличие от известных схем преобразования информации (гаммирования, маскирования сигнала), входная информационная последовательность $u(t)$ подается на вход динамической системы, и выход $y(t)$ при этом может и не зависеть явно от $u(t)$. При этом система (1), (2) порождает однозначное отображение вход – выход

$$\{x(0), u(0), u(1), \ldots\} \to \{y(0), y(1), \ldots\} \quad (3)$$

по формулам

$$y(0) = h(x(0), u(0)) = h_0(x(0), u(0)),$$

$$y(1) = h(x(1), u(1)) = h(f(x(0), u(0)), u(1)) = h_1(x(0), u(0), u(1)),$$

$$y(2) = h(x(2), u(2)) = h(f(x(1), u(1)), u(2)) = h_2(x(0), u(0), u(1)),$$

$$y(t) = h(x(t), u(t)) = \ldots = h_t(x(0), u(0), \ldots, u(t)), \quad (4)$$

при котором неизвестному начальному состоянию и значениям последовательности $\{u(0), u(1), ...\}$ соответствует известная выходная последовательность $\{y(0), y(1), ...\}$. Многие теоретические и практические задачи теории управления, связанные с определением состояния и параметров системы (1), построением обратных связей, сводятся к обращению этого отображения. Один из способов такого обращения может быть реализован с помощью обратной системы, т.е. системы вход – выход, у которой входом служит информация об $y(.)$ на некотором интервале, а выходом является функция $u(.)$.

Введем понятие относительного порядка входа для системы (1), (2). Так как значение функции $h(x(t), u(t))$ может не зависеть явно от значений $u(t)$, то аналогично и правая часть выражения $y(t+1) = h(f(x(t), u(t))$ может не содержать всех компонент вектора $u(t)$. Определим, на сколько шагов происходит задержка между информацией на входе и выходе системы (1), (2). Это величина и указывает на относительный порядок входа. Пусть $Y_t = (y(0), y(1), ..., y(t))$, $H_t = (h_0(.), h_1(.), ..., h_t(.))$, $x = x(0)$, $u = u(0)$, остальные компоненты векторов входной последовательности, содержащиеся в правых частях (4), обозначим $v_t = (u(1), u(2), ..., u(t))$. В этих обозначениях передаточное отображение (3) может быть переписано в виде

$$Y_t = H_t(x, u, v_t). \tag{5}$$

Будем говорить, что система (1), (2) имеет относительный порядок $\alpha > 0$ в некоторой области, если для всех x, u, v_t из этой области

$$\operatorname{rank}\frac{\partial H_i(x,u,v_i)}{\partial u} < m, \quad i = 0, 1, ..., \alpha-1; \quad \operatorname{rank}\frac{\partial H_\alpha(x,u,v_\alpha)}{\partial u} \equiv m.$$

Таким образом, относительный порядок α для дискретной системы указывает на номер элемента выходной последовательности, на которое явно влияют все компоненты первого элемента входной последовательности – вектора $u(0)$. В общем случае решение уравнений $Y_\alpha = H_\alpha(x, u, v_\alpha)$ относительно u имеет вид $u = H_\alpha^{-1}(Y_\alpha, x, v_\alpha) = G(Y_\alpha, x, v_\alpha)$, что не позволяет определить u без знания значений v_α. Достаточным условием того, что решение алгебраической системы (5) не зависит от v_α , является равенство [2]

$$\operatorname{rank}\frac{\partial H_i(x,u,v_\alpha)}{\partial(u,v_\alpha)} = m + \operatorname{rank}\frac{\partial H_i(x,u,v_\alpha)}{\partial v_\alpha}.$$

Подставляя выражение $u = G(Y_\alpha, x)$ в уравнения (1), получаем динамическую систему

$$x(t+1) = f(x(t), G(Y_\alpha(t), x(t))), \quad x(0) = x_0, \tag{6}$$

выход которой совпадает со входом исходной системы (1)

$$u(t) = G(Y_\alpha(t), x(t)). \tag{7}$$

Система (6), (7) является обратной динамической системой управления. Так как, по построению, $Y_\alpha(t) = (y(t), y(t+1), ..., y(t+\alpha))$, то в обратной системе в качестве входа должен присутствовать фрагмент будущих значений выхода (2). При одинаковых начальных состояниях траектории исходной и обратной систем совпадают. Поэтому можно считать, что обратная система является альтернативной формой описания одного и того же отображения вход – выход (4).

Для систем со скалярным входом и выходом ($m = p = 1$) условия однозначного обращения отображения (4) заметно упрощаются. Равенства (5) в этом случае не содержат переменных v_α. При этом относительный порядок означает номер элемента выходной последовательности, на который явно влияет первый элемент входной последовательности.

Пример. Рассмотрим передатчик (шифратор) – нелинейную систему вход – выход, на вход которой подается сообщение $u(t)$

$$\begin{aligned} x_1(t+1) &= x_2(t)x_3(t) \bmod N, \\ x_2(t+1) &= x_1(t)x_3(t) + u(t) \bmod N, \\ x_3(t+1) &= x_1(t)x_2(t) \bmod N, \\ y(t) &= x_2(t). \end{aligned} \tag{8}$$

Сигнал $y(t)$ направляется в коммуникационную сеть. Ключом для расшифрования являются неизвестные начальные условия системы $x_1(0)$ и $x_3(0)$. Приемник (дешифратор) – обратная система, с помощью которой при известном ключе проводится восстановление состояния передающей системы

$$\begin{aligned} X_1(t+1) &= X_2(t)X_3(t) \bmod N, X_1(0) = x_1(0), \\ X_2(t+1) &= y(t+1), \\ X_3(t+1) &= X_2(t)X_1(t) \bmod N, X_3(0) = x_3(0). \end{aligned} \tag{9}$$

Искомое входное воздействие определяется по формуле

$$u(t) = y(t+1) - X_1(t)X_3(t) \bmod N. \tag{10}$$

2. Эффект динамической деградации

Система (8) в отсутствие входного воздействия является нелинейной, и ее траектории при больших N обладают достаточно сложным поведением. Исключением являются траектории, лежащие на инвариантных множествах: $x_i(.) = 0$ либо $x_i(.) = x_j(.)$, $i, j = 1, 2, 3$. В первом случае состояние системы уже через шаг переходит в положение равновесия – начало координат. В общем случае, траектории, попав на инвариантные множества, остаются на нем во все последующие моменты, что приводит к падению размерности фазового пространства состояний системы. При введении в правую часть последовательности $u(t)$ система (8) становится неавтономной, и естественно предполагать, что это лишь усложнит динамику выхода и тем самым повысит сложность задачи восстановления входа. Вместе с тем анализ результатов моделирования процесса передачи информации с помощью описанной схемы показывает, что, начиная с некоторого момента t, выход системы в точности совпадает с ее входом в момент t–1. Из этого следует предположение, что собственная динамика системы (8) перестает влиять на передаваемый сигнал.

Действительно, неавтономная система (8) также обладает инвариантными множествами $x_1(.) = 0$, $x_3(.) = 0$ и $x_1(.) = x_3(.)$. При этом из полученных формул следует следующее рекуррентное выражение для определения $u(t)$:

$$u(t) = y(t + 1) - C \prod_{j=0}^{t} y(j),$$

где C равно $x_1(0)$ для четных и $x_3(0)$ для нечетных значений t. Из последнего равенства, в частности, следует, что если для некоторого целого M значение выхода $x_1(M)\, x_3(M) + u(M) = 0 \bmod N$, то для любого $i > M$ имеем $y(i) = u(i–1)$. Таким образом, вместо шифрования информационной последовательности $u(i)$ выход рассматриваемой динамической системы для любых значений ключевых параметров, начиная с некоторого момента, в точности передает значение входа с единичной задержкой.

Определение. Эффект вырождения собственной динамики системы при введении в правую часть неавтономного возмущения, выраженный в виде падения размерности пространства состояний, назовем динамической деградацией.

Безусловно, при составлении схем преобразования и передачи информации с использованием динамических систем необходим учет этого эффекта. Для того чтобы избежать влияния динамической деградации, можно применить следующую схему. Поскольку динамика состояний для передающей и принимающей систем совпадает, то достаточно рассмотреть одну из них, например систему (1), (2). На первом шаге, одновременно с выбором уравнений для передатчика, требуется найти явное описание всех инвариантных множеств, допускаемых этой системой. Далее, при разработке алгоритмов передачи и приема сигнала должна быть предусмотрена проверка условий вырождения (попадания траектории на инвариантное множество). В случае такого попадания в некоторый момент t передающая и принимающая системы должны по согласованному правилу изменить последующее состояние на состояние, не принадлежащее инвариантному множеству. Тем самым траектория будет выведена, по крайней мере на какое-то число шагов, за его пределы. В частности, для рассмотренного примера может быть применено следующее правило: при наступлении в момент t одного из событий $x_i(t) = 0$, $i \in \{1, 2, 3\}$ или $x_1(t) = x_3(t)$ на следующем шаге системы (8), (9) стартуют с исходного начального условия: $x(t+1) = X(t+1) = x_0$.

3. Автоматы-аналоги

При компьютерном моделировании бесконечных динамических систем фактически осуществляется замена исходной системы некоторым конечным аналогом, сохраняющим требуемые свойства исходной системы с определенной точностью. Оставляя в стороне вопросы точности моделирования исходной системы конечной, будем рассматривать заданную систему как прототип для построения конечной системы-аналога, переход к которой может быть осуществлен различными способами. В работе выбран один из возможных вариантов такой замены исходной динамической системы конечным автоматом с достаточно большими входным, внутренним и выходным алфавитами. При этом автомат описывается системой уравнений в конечном поле или кольце, возникающих естественным образом из описания исходной системы.

Автомат понимается как пятерка объектов $A = (X, U, Y, \delta, \lambda)$ [4], где X – множество состояний, U – входной алфавит, Y – выходной алфавит, δ: $X \times U \to X$ – функция переходов, λ: $X \times U \to Y$ – функция выходов.

Автомат называется автоматом без потери информации (БПИ), если из равенства $\lambda(x, u_1) = \lambda(x, u_2)$ следует равенство $u_1 = u_2$ для любых $x \in X$, $u_1 \in U$, $u_2 \in U$. Если множества состояний, входов и выходов автомата конечны, автомат называется конечным. Из контекста будет понятно, когда рассматриваются конечные автоматы. Функции δ и λ расширяются на множество U^* слов конечной длины обычным образом.

Далее автомат удобно описывать системой уравнений над конечным кольцом или полем. В этом случае его функционирование рассматривается в дискретном времени $t \in T = \{0, 1, 2, ...,\}$ и задается каноническими уравнениями, например, в таком варианте:

$$x(t+1) = \delta(x(t), u(t)),$$
$$y(t) = \lambda(x(t), u(t)), \; t \in T. \qquad (11)$$

Оставляя в стороне вопросы приближения, возникающие при переходе от исходной системы к ее автоматной модели, принимаем во внимание лишь то, что конечность числа значений участвующих в них величин и необходимость сохранения формы уравнений, отражающей связи между этими величинами, делают естественным рассмотрение этих уравнений как уравнений в конечных полях [5] или кольцах. Поле, содержащее q элементов, обозначается через $GF(q)$. Ниже уравнения (12) дают пример такого автомата (названного автоматом Лоренца [6]), получающийся в результате перехода от непрерывной системы Лоренца как прототипа [7] и введения в нее входного воздействия:

$$\begin{aligned} x_1(t+1) &= x_1(t) + hA_1(x_2(t) - x_1(t)), \\ x_2(t+1) &= x_2(t) + h(A_2x_1(t) - x_2(t) - x_1(t)x_3(t) + Au(t)), \\ x_3(t+1) &= x_3(t) + h(x_1(t)x_2(t) - A_3x_3(t)) \\ y(t) &= x_2(t) + h(A_2x_1(t) - x_2(t) - x_1(t)x_3(t) + Au(t)). \end{aligned} \tag{12}$$

Уравнения (13) описывают обратный автомат:

$$\begin{aligned} X_1(t+1) &= X_1(t) + hA_1X_2(t) - X_1(t)), \\ X_2(t+1) &= y(t+1), \\ X_3(t+1) &= X_3(t) + h(X_1(t)X_2(t) - A_3X_3(t)), \\ u(t) &= (((y(t) - X_2(t))\cdot h^{-1} - A_2X_1(t) + X_2(t) + X_1(t)X_3(t))\cdot A^{-1}. \end{aligned} \tag{13}$$

Далее все операции в уравнениях понимаются как операции в некотором поле $GF(q)$. Заметим, что входной сигнал в систему можно вводить разными способами, выполняя лишь требование обратимости системы. В автоматном случае это означает, что автомат-преобразователь должен быть БПИ-автоматом [4] или без потери информации конечного порядка, если допускается обращение системы с запаздыванием. Автоматы Лоренца, как легко видеть, являются БПИ-автоматами в любом поле. Так как обычно речь идет об обработке информации с помощью компьютеров, то такая информация представляется последовательностью битов, более крупных единиц – байтов или блоков, кратных байтам по длине. В этом случае число различных элементов, описываемых всевозможными комбинациями значений отдельных битов, равно $2^m = q$, где $m = 8k$, $k \in N$. Поэтому соответствующие вычисления можно проводить либо в кольце Z_q, либо в поле $GF(2^m)$. Так как компьютерная обработка информации осуществляется побайтно, то реализацию автоматных аналогов удобно рассматривать в полях $GF(2^{8k})$, $k = 1, 2, ...$ В этом случаев в качестве базового рассматривается поле $GF(q) = GF(2^8)$ и неделимым элементом информации выступает байт. Такое поле строится как кольцо классов вычетов многочленов над полем $GF(2)$ по неприводимому над этим полем многочлену, например такому: $f(x) = x^8 + x^4 + x^3 + x^2 + 1$. Вычеты $A_0 + A_1x + A_2x^2 + A_3x^3 + A_4x^4 + A_5x^5 + A_6x^6 + A_7x^7$ по модулю соответствующего многочлена описываются как булевы векторы A_0, A_1, ..., A_7, где A_i равно 0 или 1, $i = 0, 1, ..., 7$. Эксперименты по шифрованию информации с помощью автоматов Лоренца позволили обнаружить следующий эффект. Если входное слово, подаваемое на автомат, подвергается искажениям (например, один из символов меняется на какой-то другой), то возможны два варианта: либо выходное слово, начиная с момента искажения, полностью изменяется, либо через некоторое число шагов после момента искажения оно совпадает с соответствующим конечным отрезком неискаженного выходного слова. Последнее свойство аналогично свойству самосинхронизируемости некоторых поточных шифрсистем [8] и говорит об определенной устойчивости автомата к искажениям входной информации. Этот эффект и наблюдался при расшифровывании искаженной информации автоматами Лоренца. В [6] на основе введенных понятий синхронизируемости и k-локальной синхронизируемости состояний описана структура некоторых автоматов, обладающих такими свойствами. Способность восстановления функционирования алгоритма преобразования после искажения входных последовательностей может либо поддерживаться (как, например, в самосинхронизирующихся поточных шифросистемах), либо подавляться, как свойство, ограничивающее распространение искажения одного символа на возможно большее число символов шифртекста. Такие особенности могут влиять на криптостойкость алгоритмов. В п. 2 описано явление деградации, связанное с вырождением множества траекторий динамической системы в результате попадания в некоторое подмножество состояний фазового пространства. В [6] показано, что в автоматном случае это свойство может быть следствием особенностей структуры графа переходов автомата, которую можно описать следующим образом. На множестве состояний автомата определяется специальная конгруэнция k-локальной синхронизируемости состояний, которая описывает свойство «устойчивости» к «искажениям» подслов фиксированной длины во входных последовательностях. Доказано, что факторизация по этой конгруэнции определяет фактор-автомат, в котором множество состояний распадается на непересекающиеся циклы. Уход с этих циклов возможен только при дополнительных управляющих воздействиях на автомат, переводящих его в новое состояние. Такое состояние, вообще говоря, может и не принадлежать исходному множеству состояний. Далее предлагается способ формирования таких воздействий, заключающийся в регулировании размерности пространства состояний и изменении, таким образом, исходного множества состояний.

4. Управление размерностью пространства состояний

Пусть $A = (X, U, Y, \delta, \lambda)$ – автомат Мили. Автомат $B = (X_B, U_B, Y_B, \delta_B, \lambda_B)$ будем называть подавтоматом автомата A (и писать $B \subseteq A$), если $X_B \subseteq X$, $U_B \subseteq U$, $Y_B \subseteq Y$, а δ_B и λ_B есть сужения соответственно функций δ и λ на множество $X_B \times U_B$.

Пусть автомат A описывается системой (11) над полем $GF(q)$. В этом случае автомат будем обозначать через A_q. Результаты теории конечных полей позволяют считать, что $GF(q)$ есть подполе поля $GF(q^n)$ при любом натуральном n. В силу этого уравнения (11) можно понимать как уравнения, задающие некоторый автомат A_q^n в поле $GF(q^n)$. Ограничения его функций на поле $GF(q)$, в силу замкнутости последнего, определяют подавтомат, изоморфный автомату A_q, который будем обозначать таким же образом. Пусть задана последовательность расширений поля $GF(q)$, $GF(q^{m_1})$, $GF(q^{m_2})$, …, $GF(q^{m_n})$, такая, что $m_i | m_{i+1}$, $i = 1, \ldots, n–1$. В этом случае $GF(q) \subset GF(q^{m_1}) \subset GF(q^{m_2}) \subset \ldots \subset GF(q^{m_n})$, что определяет, в силу вышесказанного, последовательность автоматов $A_q \subset A_q^{m_1} \subset \ldots \subset A_q^{m_n}$. Если же числа m_i, $i = 1, \ldots, n–1$, произвольны (в частности, попарно взаимно просты), то поле $GF(q^m)$, где m – наименьшее общее кратное этих чисел, содержит всякое подполе $GF(q^{m_i})$, а значит, имеется и вложение соответствующих автоматов. Более того, все сказанное справедливо и в случае, когда в качестве расширения поля $GF(q)$ выбирается его алгебраическое замыкание (обозначим его $GF(q^\infty)$). Тогда и соответствующие уравнения, задающие исходный автомат, можно понимать как уравнения в поле $GF(q^\infty)$, и эти уравнения задают уже бесконечный автомат A_q^∞. Из единственности алгебраического замыкания $GF(q^\infty)$ и вышесказанного следует

Утверждение. Для произвольных автомата A_q и натурального m справедливы включения $A_q \subset A_q^m \subset A_q^\infty$.

Сказанное обосновывает построение нового автомата $A(m_1, \ldots, m_n)$ фактически с переменными множествами состояний, входов и выходов. Его функционирование в любой момент времени совпадает с функционированием одного из автоматов $A_q^{m_i}$, $i = 1, \ldots, n$. Смена одного автомата другим или «принудительная» смена текущего состояния в процессе функционирования может осуществляться при выполнении некоторого предиката $P(x, p)$, определенного на $X \times U^*$, где X и U – множества состояний и входов соответственно текущего автомата A.

В качестве указанного предиката можно выбрать, например, условие появления деградации, описанное выше, условие попадания в определенные состояния или появление фиксированных подслов во входной последовательности. Например, для систем (8), (9) такой предикат можно определить как $P = ((x_1(t) = x_3(t)) \vee (x_1(t) \cdot x_2(t) \cdot x_3(t) = 0)) \wedge (u(t) \cdot u(t–1) \cdot u(t–2) = 0))$. Его истинностное значение определяет «принудительную» смену состояния, например, по такому правилу: $x_1(t+1) = u(t)$, $x_2(t+1) = u(t–1)$, $x_3(t+1)) = u(t–2)$, либо, если отказаться от условия неравенства нулю трех последовательных входных символов, по более простому правилу из п. 2: $x(t+1) = X(t+1) = x_0$.

Дополнительные вычисления для проверки истинности предиката, естественно, повышают сложность алгоритма преобразования информации. Это требует наложения ограничений на временную и емкостную сложности вычисления таких предикатов. В приведенном примере эту сложность легко оценить: при фиксации верхней границы размерности обе сложности оцениваются некоторыми константами. В целом это не повышает порядка оценок по сравнению с алгоритмом без вычисления предиката. В общем случае ситуация более сложна и требует исследования структуры автомата.

Пусть входная последовательность $p = u_1 u_2 \ldots u_k$ в некотором исходном алфавите $U = GF(q)$ подается на автомат $A(m_1, \ldots, m_n)$. Она обрабатывается этим автоматом либо посимвольно, либо блоками размера m_i, $i = 1, \ldots, n$, где размер блока определяется выбором поля, как указывалось выше. В результате обработки всей последовательности p она оказывается разбитой на подслова разной длины (блоки), каждое из которых преобразуется своим автоматом. Это разбиение заранее неизвестно и определяется предикатом $P(x, p)$. Если он существенно зависит от x, этот параметр или иной другой можно сделать секретным, в дополнение к секретным коэффициентам, задающим конкретный автомат-шифратор в каждом сеансе преобразования. В качестве такого предиката для автоматов Лоренца, учитывая возможность попадания на вышеуказанные циклы фактор-автомата, может быть выбрано, например, условие совпадения текущего состояния с заранее заданным состоянием, уже появлявшимся в один из предыдущих моментов времени, либо условие появления во входном слове заданного подслова. При выполнении этих условий происходит изменение размерности очередного обрабатываемого блока. Размер блока может выбираться из заранее оговоренного списка значений в порядке, который либо жестко фиксирован, либо снова может определяться некоторым предикатом. Если в списке все значения размерностей взаимно просты, то наименьшая размерность поля (как векторного пространства), содержащего все подполя выбранных размерностей, равна произведению этих размерностей. Это усложнит анализ поведения шифрующего автомата на основе выбора, в силу утверждения 1, в качестве исходного поля указанного надполя. Помимо этого, усложнение анализа поведения шифрующего автомата определяется также тем, что задача восстановления разбиения входного слова на заданные подслова, каждое из которых обрабатывается своим подавтоматом, относится к классу задач упаковки. Эта задача может решаться перебором, если заранее известен список возможных размерностей. Однако прямой перебор затруд-

нен, так как число вариантов указанных разбиений с ростом длины слова даже при небольших значениях m_i растет достаточно быстро.

Заключение

Предложенный метод динамического изменения размерности пространства состояний позволяет усложнять поведение системы без принципиального изменения ее описания и сложности алгоритма преобразования. Пополнение описания системы условиями, меняющими размерность пространства состояний, а не саму систему уравнений означает, что фактически таким образом вводятся, помимо параметрических, структурные ключи, которые также могут быть секретными. Это позволяет повысить стойкость соответствующих алгоритмов преобразования информации. Предложенный вариант обработки информации динамическими системами с переменной размерностью пространства состояний может быть полезным при разработке шифров с регулируемой степенью криптостойкости. При этом необходимы дополнительные исследования, связанные с вопросами формирования последовательностей примитивных полиномов для описания соответствующих полей и автоматов, с оценкой усложнения соответствующих алгоритмов и по памяти, и по времени, и др.

ЛИТЕРАТУРА

1. *Feldmann U.*, *Hasler M. and Schwarz W.* Communication by chaotic signals: the inverse system approach // Int. J. Circ. Theory Appl. 1996. V. 24. P. 551 – 579.
2. *Ковалев А.М.*, *Щербак В.Ф.* Управляемость, наблюдаемость, идентифицируемость динамических систем. Киев: Наукова думка, 1993. 285 с.
3. *Ковалев А.М.* Критерии функциональной управляемости и обратимости нелинейных систем // ПММ. 1998. Т. 62. Вып. 1. С. 110 – 120.
4. *Кудрявцев В.Б.*, *Алешин С.В.*, *Подколзин А.С.* Введение в теорию автоматов. М.: Наука, 1985. 320 с.
5. *Лидл Р.*, *Нидеррайтер Т.* Конечные поля. М.: Мир, 1988. Т. 1, 2. 820 с.
6. *Козловский В.А.*, *Толмачевская Л.А.* Автоматные аналоги динамических хаотических систем // Труды ИПММ НАН Украины. 2003. Т. 8. С. 59 – 69.
7. *Данилов Ю.А.* Лекции по нелинейной динамике. Элементарное введение. М.: Постмаркет, 2001. 184 с.
8. *Алферов А.П.*, *Зубов А.Ю.*, *Кузьмин А.С.*, *Черемушкин А.В.* Основы криптографии. М.: Гелиос АРВ, 2002. 480 с.

2008 Математические методы криптографии № 2(2)
DOI 10.17223/20710410/2/10
УДК 004.4

О ПРЯМОМ ОПЕРАЦИОННОМ АНАЛИЗЕ СИММЕТРИЧНЫХ ШИФРОВ

А.М. Кукарцев, А.М. Попов, В.С. Шестаков

Сибирский государственный аэрокосмический университет им. академика М.Ф. Решетнева, г. Красноярск

E-mail: amkukarcev@mail.ru, alexey_m_popov@newmail.ru, shockk@mail.ru

Рассматривается методика исследования криптографических алгоритмов и их ключей с позиции их математического аппарата. Уточняется понятие сложности криптоалгоритма как относительной оценки криптостойкости. Описывается реализация метода прямого операционного анализа.

Ключевые слова: *криптография*, *композиционные криптоалгоритмы*, *анализ криптостойкости.*

Проблема безопасности криптоалгоритмов является многогранной. В ней можно выделить организационную и техническую стороны. Ошибки могут быть заложены и в самом алгоритме. Даже алгоритмы, которые признаны стойкими, при неправильном использовании будут слабыми. Ещё более важным является правильное использование ключевой информации. Практически к каждому алгоритму поставляется перечень заведомо слабых ключей. Но слабость таких ключей заключается в том, что они не стойки к уже известным методам криптоанализа. Наибольшую актуальность проблема стойкости ключей приобретает в алгоритмах, для которых требуется частая плановая их смена. Возможно ли проанализировать алгоритм перед его использованием? Ещё более интересен вопрос: а можно ли проектировать заведомо безопасный шифр (либо дать исчерпывающий перечень рекомендаций по его безопасному использованию) и его (что немаловажно) реализацию? Ведь каждый новый алгоритм шифрования требует детального и тщательного изучения. В настоящей работе рассматривается методика детального исследования некоторых криптографических алгоритмов.

1. Метод операционного анализа и изоалгоритмы

Для любого криптоалгоритма можно определить изоморфизм, который ставит в соответствие объектам, преобразуемым алгоритмом, булевы векторы. В качестве отображения можно использовать перевод значений из различных позиционных систем счисления в двоичную. Используя изоморфизм, можно перейти от алгоритма как последовательности операций к системе булевых функций и вести анализ последней [1].

Детальное исследование полученных булевых функций позволяет анализировать основную составляющую криптостойкости – математическую сложность самих алгоритмов и их ключей. Цель данной работы – описать теоретическую часть этой методики, уточнить понятие сложности алгоритмов, реализовать средства анализа и апробировать методику на шифре простой замены – сложении чисел в двоичном позиционном коде.

Так как сам метод анализа является вычислительно сложным, то следует также рассмотреть и недостатки методики. Такие недостатки возникают при её практической реализации. В работе рассмотрены варианты, позволяющие реализовать описываемую методику анализа.

Для начала опишем основные понятия, применяемые для компактности изложения метода. На рис. 1 показана модель «чёрного» ящика рассматриваемых преобразований.

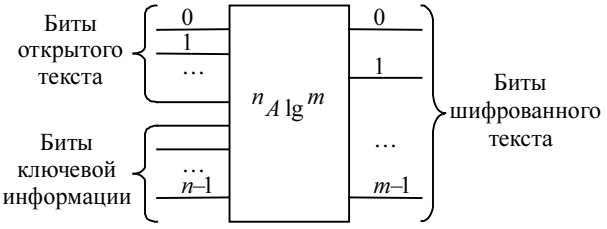

Рис. 1. «Чёрный ящик» алгоритмов шифрования

Определим следующие понятия, применяемые при описании метода. Далее будем называть *аргументом-вектором* булев вектор, который является элементом множества, изоморфного множеству входных объек-

тов. Будем называть *альфа-вектором* булев вектор, который является элементом множества, изоморфного множеству выходных объектов. Будем называть *изоалгоритмом* и обозначать ${}^{n}A\,\mathbf{lg}^{m}$ систему булевых функций, отображающих аргумент-вектор в альфа-вектор:

$$ {}^{n}A\,\mathbf{lg}^{m}:\begin{cases} y_0 = f_0(x_{n-1}, x_{n-2}, \ldots, x_0), \\ y_1 = f_1(x_{n-1}, x_{n-2}, \ldots, x_0), \\ \quad\vdots \\ y_{m-1} = f_{m-1}(x_{n-1}, x_{n-2}, \ldots, x_0). \end{cases} \tag{1} $$

Так как все изоалгоритмы имеют одинаковую математическую природу (системы булевых функций), то возможно ввести общие характеристики, по которым можно будет сравнивать различные исходные алгоритмы.

Для вычисления значения конкретного бита зашифрованного текста необходимо проделать какое-то количество логических операций, которые будут отражать вычисление значения булевой функции. Для различных булевых функций такое количество операций различно. С позиции анализа «грубой силы» необходимо минимальное количество операций для перебора возможных вариантов. Так как функции булевы, то такими операциями являются функции аристотелева базиса.

Булева функция может быть представлена в виде двух нормальных форм – дизъюнктивной и конъюнктивной. Тогда определим сложность как минимально необходимое количество логических операций для преобразования вектора-аргумента в компоненту альфа-вектора среди всех возможных КНФ и ДНФ. Единицей измерения считается операция на бит, оп/бит (operation per bit, opb).

Под сложностью всего изоалгоритма будем понимать среднее значение сложности всех булевых функций, входящих в его состав. Можно считать сложностью изоалгоритма вектор сложностей булевых функций. Так как важно значение каждого бита, то возможно проработать более эффективную методику расчёта сложности для всего изоалгоритма. Метод прямого операционного анализа основывается на непосредственном переходе от таблицы булевых векторов к системе булевых функций (1).

2. Интерпретация значений сложности

Расчёт значений сложности является технически трудной задачей. Но даже если такие значения получены, то встаёт вопрос о том, что они выражают. Проведённые аналитические исследования и практическая проверка показали, что среди форм булевых функций можно выделить идеальные нормальные формы.

Идеальная нормальная форма (*ИНФ*) – это совершенная нормальная форма, которая также является минимальной нормальной формой. Такую форму имеют, например, симметрические разности различного числа переменных.

Есть предположение, что ИНФ имеют максимальную сложность среди всех остальных булевых функций той же разрядности. Это позволяет определить *верхнюю границу* сложности булевой функции от n переменных как

$$ d_{\max} = 2^{n-1}(n+1). \tag{2} $$

Минимальную границу сложности считаем равной нулю. Такой сложности соответствуют булевы функции, тождественно равные константам (0 или 1).

Для криптографии максимальная сложность (2) не интересна для рассмотрения, так как важна сложность алгоритма с фиксированным ключом, и если алгоритм с фиксированным ключом будет иметь максимальную сложность, то ему будут соответствовать очень мало различных функций. В этом случае при известном преобразовании, тем более в форме булевых функций, построить обратное, не зная ключа, не составляет труда. Малая сложность также плоха, так как не маскирует зависимость выходных бит от входных. Таким образом, можно принять, что *оптимальная сложность* находится посредине между максимальной и минимальной и определяется как

$$ d_{\text{opt}} = 2^{n-2}(n+1). \tag{3} $$

3. Прямой операционный анализ шифра простой замены – сложения чисел заданной разрядности

В качестве примера операционного анализа алгоритмов преобразований рассмотрим алгоритм классического поразрядного сложения с переносом. Тогда под вектором-аргументом будем понимать пару слагаемых, причём первая половина соответствует первому слагаемому, вторая – второму (ключу). Старшинство бит в половинах вектора-аргумента соответствует старшинству бит в слагаемом. Так как присутствует единица переноса, то длина альфа-вектора будет равна половине длины аргумента-вектора плюс единица.

Для эксперимента возьмём длины векторов аргументов 2, 4, 6, 8 и 10 бит. Таким длинам соответствуют алгоритмы сложения: ${}^{1+1}A\,\mathbf{lg}^{1+1}_{(\mathbf{mod}\,2)}$, ${}^{2+2}A\,\mathbf{lg}^{2+1}_{(\mathbf{mod}\,4)}$, ${}^{3+3}A\,\mathbf{lg}^{3+1}_{(\mathbf{mod}\,8)}$, ${}^{4+4}A\,\mathbf{lg}^{4+1}_{(\mathbf{mod}\,16)}$ и ${}^{5+5}A\,\mathbf{lg}^{5+1}_{(\mathbf{mod}\,32)}$. В значениях

длин аргументов-векторов выделена ключевая и информационная часть, в значении длин альфа-векторов – единица переноса.

Также проанализируем изменение сложности при фиксации ключевых бит (указание ключа). Это позволит анализировать слабые и сильные ключи.

Важно отметить, что изоалгоритм, соответствующий алгоритму шифрования и имеющий сложность, близкую к оптимальной, не может считаться безопасным. Это связано с тем, что при фиксации бит ключа сложность может изменяться достаточно сильно. Таким образом, если рассматривать алгоритм совместно с конкретным ключом, то ключ можно считать настройкой алгоритма. Ключ перестраивает внутренние связи алгоритма, и соответственно получается преобразователь с длиной вектора-аргумента соответствующей разности общей длины и ключа. По этой же длине необходимо оценивать верхнюю и оптимальную границы сложности.

На базе компонентов CryptoLab была разработана утилита operationAnalyzerSum, которая анализирует сложение. Данная утилита базируется на системе классов CryptoLab. Система классов CryptoLab содержит расширение для средств операционного анализа. Эти средства представлены реализацией алгоритма «Вариационное дерево»[1], алгоритма расчёта сложности и алгоритма фиксации ключевых бит.

Общее время работы утилиты при расчёте сложностей с возможностью фиксации ключа составило 27 мин на машине класса PC AMD Athlon 2500+, RAM 512 Мбайт, Microsoft Windows XP Professional выпуска апреля 2005 г. Исходные данные анализа – значение n. Утилита последовательно перебирает все целые значения от 2 до n включительно. Утилита формирует также основной протокол работы и выводит данные об изоалгоритмах в промежуточные файлы. Создаются как текстовые файлы, так и бинарные. Результаты работы утилиты показаны на графиках.

На рис. 2 представлены значения сложности алгоритма шифрования, на рис. 3 – значения средней сложности по ключам.

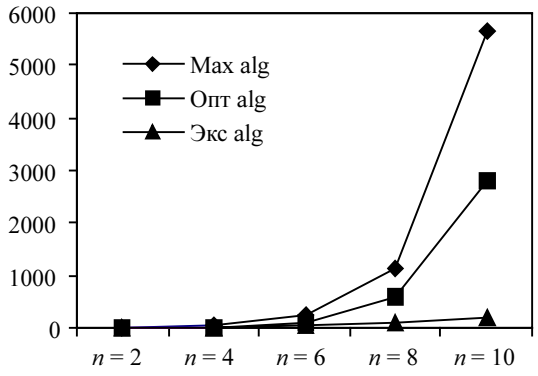

Рис. 2. Значения сложности алгоритма

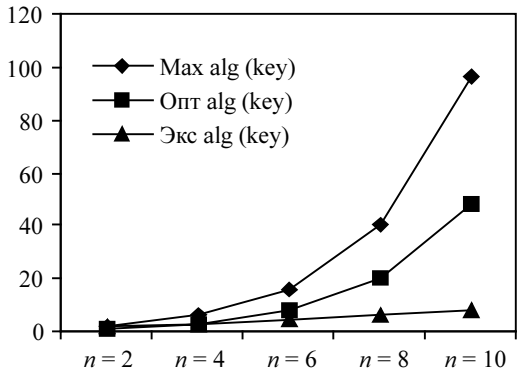

Рис. 3. Средние значения сложности алгоритма с фиксированными ключами

Для проведения анализа сложности с фиксированным ключом использовались средние оценки. Возможность применения средних оценок объясняется тем, что если присутствуют ключи, дающие требуемое значение сложности, но их мало (в процентном отношении к общему числу ключей), то их нельзя использовать. Это связано с тем, что криптоаналитик, скорее всего, сначала проверит именно очень сильные (сложность, близкая к максимальной) ключи и очень слабые ключи (сложность, близкая к нулю). Таким образом, интересно большое множество ключей, которые и попадают под среднюю оценку.

Из рис. 2 и 3 можно сделать вывод о том, что при увеличении длины вектора-аргумента средняя сложность изоалгоритма как с ключом, так и без него резко падает по сравнению с оптимальной границей. Средняя сложность с ростом длины аргумента-вектора растёт почти линейно, в отличие от оптимальной границы.

Таким образом, сложность можно использовать для анализа криптографических алгоритмов и производить следующие сравнения и оценки:

- сопоставлять различные алгоритмы с различной длиной ключа;
- сопоставлять различные ключи в рамках одного алгоритма;
- оценка сложности в зависимости от длины ключа;
- оценка эффективности использования композиций алгоритмов.

Актуальна, однако, проблема возможности анализа для больших длин векторов-аргументов. Так, например, утилита, использованная в эксперименте, не закончила анализ исходного преобразования с длиной вектора-аргумента, равной 12 бит, за сутки.

4. Возможность операционного анализа

Операционный анализ позволяет исследовать достаточно большой класс алгоритмов. Но тогда зададимся вопросом о том, насколько возможен в принципе такой анализ.

Рассмотрим методику расчёта сложности. Так как для расчёта сложности необходима система булевых функций (1), то необходимы все значения альфа-векторов. Определим соотношения, связывающие количество строк таблицы альфа-векторов, полный её объём и примерное время для её построения. Пусть τ_0 – время, необходимое для преобразования одного аргумента-вектора в альфа-вектор (фактически, это есть время зашифрования одного блока открытого текста), τ – время построения таблицы, c – количество строк в таблице, V – полный объём таблицы в битах. Тогда

$$c = 2^n, \ \tau = c\tau_0 = 2^n\tau_0. \tag{4}$$

Если принять во внимание, что перебираются все значения аргумента-вектора, то можно упорядочить таблицу и убрать из неё данные об аргументе-векторе. Тогда объём таблицы будет

$$V = cm = 2^n m. \tag{5}$$

Так, для DES по (5) имеем $V = 2^{120} \times 2^6 = 2^{126}$ бит. То есть расчёт сложности для DES технически невозможен уже потому, что негде хранить экспериментальный материал, необходимый для анализа. Даже если его удастся разместить, то время, необходимое на генерацию данных по (4) при условии, что шифрование одного блока реализуется одной (хотя в действительности их несколько сотен) операцией для процессора на частоте 2,5 ГГц, составит $\tau = 2^{120}/(2{,}5\cdot 10^9\ \text{Гц}) \approx 10^{26}$ с $> 10^{18}$ лет.

Будем называть *прямым операционным анализом* расчёт сложности через построение таблицы альфа-векторов. Любой алгоритм с шириной входа больше 50 бит технически не может быть подвергнут прямому операционному анализу. Следовательно, необходимы пути обхода этой границы.

Большинство алгоритмов шифрования не являются монолитными преобразованиями и, следовательно, могут быть представлены как композиции более мелких (с меньшей сложностью и меньшими длинами аргументов-векторов и альфа-векторов) преобразований. Так, например, DES может быть представлен как композиция преобразований, соответствующих одному раунду. В свою очередь, раунд DES также может быть представлен как композиция более мелких преобразований – сумм, сдвигов, таблиц подстановки и перестановки, симметрической разности. Аналогично может быть представлен алгоритм шифрования ГОСТ 28147-89.

Разбиение алгоритма на блоки удобно делать, если получаются преобразования со сравнительно малой длиной вектора-аргумента. Здесь и далее под длиной будем понимать длину вектора-аргумента, так как многие технические характеристики зависят экспоненциально от длины вектора-аргумента, а от длины альфа-вектора линейно. Например, в алгоритмах DES и ГОСТ 28147-89 длины S-блоков будут равны соответственно 6 и 4 бит. Преобразования такой длины уже поддаются прямому операционному анализу.

Изоалгоритмы некоторых преобразований могут быть получены непосредственно из их записи. В частности, такими преобразованиями являются таблицы перестановки бит. Таблицы перестановки в большинстве случаев используются в симметричных блочных криптоалгоритмах (DES, AES, ГОСТ 28147-89). Фактически, булевы функции таких преобразований не будут содержать никаких операций, а будут только присваивать функции тот или иной компонент вектора-аргумента. Будем называть *транспозиционным операционным анализом* расчёт сложности, который производится в булевых функциях, полученных из таблиц перестановок.

Изоалгоритмы, соответствующие некоторым элементарным преобразованиям для больших длин, могут быть получены путём экстраполяции тех же преобразований с малыми длинами. Так, например, сложение 4-битных чисел с единицей переноса может быть получено из сложения 2-битных с единицей переноса по классической пирамидальной схеме соединения сумматоров [2, с. 332]. Отдельно стоит выделить побитовые операции, например побитовую симметрическую разность. Булевы функции таких преобразований полностью повторяют друг друга с точностью до индексов компонент вектора-аргумента. Будем называть *экстраполяционным операционным анализом* расчёт сложности, который производится в булевых функциях, полученных из аналогичных булевых функций малой длины.

Если хранить булевы функции как композицию ИНФ и выполнять все операции в такой форме, то проблема хранения и обработки решается. Это позволяет в ряде случаев осуществлять операционный анализ практически.

Выводы

1. Произведено уточнение понятий сложности криптоалгоритма и операционного анализа.
2. Определены границы и формулы расчёта минимальной, максимальной и оптимальной сложности.
3. Произведён прямой операционный анализ шифра простой замены – сложения по модулю степени 2 заданной разрядности. Представлены значения сложности алгоритма, в том числе с фиксированными ключами.

4. Показана практическая невозможность прямого операционного анализа произвольного преобразования.

5. Введено понятие идеальной нормальной формы, позволяющее обойти алгоритмическую трудность прямого анализа частными методами – транспозиционными, экстраполяционными и композиционными.

ЛИТЕРАТУРА

1. *Кукарцев А.М.*, *Старовойтов С.А.*, *Шестаков В.С.* Операционный анализ криптоалгоритмов // Актуальные проблемы безопасности информационных технологий: Материалы I Междунар. науч.-практич. конф. (Красноярск, 27 марта 2007 г.) / Под общ. ред. О.Н. Жданова, В.В. Золоторёва. Красноярск: Сиб. гос. аэрокосмич. ун-т, 2007. С. 66 – 72.
2. *Титце У.*, *Шенк К.* Полупроводниковая схемотехника: Справочное руководство. М.: Мир, 1982. 512 с.

2008 Математические методы криптографии № 2(2)
DOI 10.17223/20710410/2/11
УДК 519.7

СОВЕРШЕННЫЕ СХЕМЫ РАЗДЕЛЕНИЯ СЕКРЕТА

Н.Г. Парватов

Томский государственный университет

E-mail: parvatov@mail.tsu.ru

В статье излагаются некоторые известные результаты теории разделения секрета.

***Ключевые слова*:** *совершенная схема разделения секрета*, *структура доступа*, *идеальное разделение секрета*, *полиматроид*, *матроид.*

В теории разделения секрета рассматривается задача, которую неформально можно поставить следующим образом. Требуется разделить значение секрета из некоторого множества секретов между участниками из некоторого множества участников, выдав каждому участнику его долю секрета так, чтобы заранее определённые, авторизованные, множества участников могли, соединив свои доли, вычислить истинное значение секрета, а участники остальных, неавторизованных, множеств не могли бы этого сделать. Иногда требуется, чтобы участники неавторизованных множеств, пытаясь восстановить истинное значение секрета, не могли бы исключить ни одного значения из множества секретов, или даже не получили бы никакой дополнительной информации о секрете. Данная задача может возникнуть при разделении доступа между группой лиц, не доверяющих друг другу, а также при хранении конфиденциальной информации, разделённой на части. Несмотря на то, что в настоящее время имеется довольно обширный перечень публикаций по разделению секрета, некоторые ключевые статьи достать довольно трудно. Столкнувшись с этой трудностью, автор вынужден был самостоятельно восстанавливать систему первоначальных понятий теории разделения секрета и получать некоторые основные результаты этой теории. О некоторых из этих результатов и пойдёт речь далее.

Структура доступа

Структурой доступа (сд) на множестве Q станем называть всякое подмножество G системы $B(Q)$ всех подмножеств множества Q, обладающее свойством монотонности:

«если часть множества $A \subseteq Q$ принадлежит G, то и само A принадлежит G».

Множества, принадлежащие сд, станем называть *авторизованными* в ней. Система всех минимальных по включению авторизованных множеств сд G называется её *базисом* и обозначается G_0. Элементы множества Q называются *участниками* сд G. Участники, принадлежащие базисным множествам из G_0, называются *существенными*. Пользуясь случаем, введём несколько важных операций над сд. Во-первых, заметим, что для любого множества U из $B(Q)$ множество $B(Q\backslash U)\cap G$ является сд на множестве участников $Q\backslash U$; эту сд станем обозначать через $G_{Q\backslash U}$. Множество всех подмножеств V из $B(Q)$, таких, что $U\cup V$ авторизованно в G, образует сд на Q. Эту сд обозначим через $G^{(U)}$. Наконец, через $G^{Q\backslash U}$ станем обозначать сд $G^{(U)}{}_{Q\backslash U}$, полученную повторным применением первых двух операций. Из данных трёх операций первая и последняя имеют особое значение и называются операциями *взятия миноров*. Сд, возникающие из данной сд G в результате (возможно, неоднократного) применения подобных операций, называются *минорами* G. Отметим, что многократное взятие миноров можно всегда заменить равносильным не более, чем двукратным применением подобных операций. Если все участники, принадлежащие множеству U, несущественны в сд G, то миноры $G_{Q\backslash U}$ и $G^{Q\backslash U}$ совпадают. В этой ситуации будем говорить, что сд G получена из сд $G_{Q\backslash U}$ (равносильно из $G^{Q\backslash U}$) путём *введения* несущественных участников, а сд $G_{Q\backslash U}$ (равносильно сд $G^{Q\backslash U}$) получена из сд G путём *удаления* несущественных участников. Классы сд, замкнутые операциями взятия миноров, а также операциями введения и удаления несущественных участников, станем называть *инвариантными*. Далее, для сд G на множестве Q через G^* принято обозначать определённую на том же множестве Q сд, в которой авторизованными являются всевозможные дополнения неавторизованных в G множеств. Ясно, что $G^{**}=G$. Сд G и G^* по отношению друг к другу принято называть *двойственными*.

В дальнейшем полезно иметь в виду, что существует взаимно однозначное соответствие

$$x \mapsto U(x)$$

между булевыми векторами $x = (x_1, ..., x_n)$ длины n и всевозможными подмножествами $U(x)$ n-элементного множества $Q = \{p_1, ..., p_n\}$, при котором $U(x) = \{p_i \mid x_i = 1\}$. Данный факт влечёт существование (обратных по отношению друг к другу) взаимно однозначных соответствий

$$f \mapsto G(f),\ G \mapsto f(G)$$

между булевыми функциями f, зависящими от n аргументов $x_1, ..., x_n$, и всевозможными системами G подмножеств множества Q. При этом система $G(f)$ состоит из всевозможных таких подмножеств $U(x)$, для которых $f(x) = 1$, а булева функция $f(G)$ принимает значение 1 на наборе x, если $U(x)$ принадлежит G. Данные соответствия, в свою очередь, индуцируют взаимно однозначное соответствие между монотонными булевыми функциями от n аргументов и структурами доступа на n-элементном множестве. Отметим, что существенность участника p_i из Q в структуре доступа G равносильна тому, что функция $f(G)$ зависит от переменной x_i существенно. Отметим также, что функции $f(G_{Q\setminus U})$ и $f(G^{Q\setminus U})$ могут быть получены из функции $f(G)$ подстановкой константы 0 и 1 соответственно на места переменных x_i, $p_i \in U$. В свете сказанного легко понятна связь между инвариантными классами структур доступа и инвариантными классами булевых функций (замкнутыми операциями перестановки переменных, операциями введения и удаления фиктивных переменных, а также операциями подстановки 0 и 1 на места переменных [1]).

Схема разделения секрета

Договоримся обозначать через A^Q множество всевозможных функций $s{:}Q\to A$. Всякую такую функцию будем называть также набором, а её значение $s(q)$ будем также обозначать через s_q и называть значением q-координаты набора s. Для любого множества U из $B(Q)$ через s_U станем обозначать ограничение $s_U{:}U\to A$, такое, что $s_U(q) = s(q)$ для всех q из U. В дальнейшем нас будут интересовать подмножества множества A^Q. Для всякого такого подмножества S станем обозначать через S_q множество всех q-координат s_q и через S_U множество всех ограничений s_U для всевозможных наборов s из S. Для набора y из S_U через S^y обозначим множество всех таких наборов s из S, что $s_U = y$. Пусть $Q = P \cup \{D\}$ и элемент D не принадлежит множеству P. Набор (S, P, D), где S – непустое подмножество множества A^Q, называется *схемой разделения секрета* (*срс*). При этом множество Q называется *множеством участников* данной срс. Участник D называется *дилером*. Множество S_D называется *множеством секретов*, число $|S_D|$ его элементов называется *порядком* срс. Множество S_q называется *множеством долей* участника q из P. Наборы из S называются *правилами разделения секрета*. Так, s из S – это правило разделения секрета s_D, а s_q – принадлежащая участнику q доля секрета s_D, разделённого по правилу s. Несложно понять, что всевозможные множества U из $B(P)$, для которых выполняется равенство

$$|S_U \cup {}_{\{D\}}| = |S_U|,$$

образуют сд, обозначим её через $G(S, P)$, на множестве P. Это – сд, *реализуемая* срс S. В соответствии с определением для любого авторизованного в $G(S, P)$ множества U и любого правила s из S значение s_D однозначно определяется ограничением s_U. Это свойство позволяет использовать срс для решения описанной во введении задачи разделения секрета. Именно для того, чтобы разделить значение секрета из множества S_D между участниками из множества P так, чтобы авторизованные в $G(S, P)$ множества участников могли восстановить секрет, нужно прибегнуть к помощи ещё одного участника, пользующегося всеобщим доверием. Этот участник выбирает наугад (или в соответствии с некоторым распределением) правило s из S и каждому участнику q из P в тайне от остальных передаёт долю s_q. Отметим, что в описанной ситуации доли участников из неавторизованного множества могут содержать некоторую (возможно, значительную или даже полную) информацию о секрете. Чтобы избавить срс от этого нежелательного свойства, понадобятся более сложные определения, которые будут сделаны далее.

Энтропийная функция

Для непустого подмножества S множества A^Q функцию $h{:}B(Q)\to[0,+\infty)$ будем называть *энтропийной*, если выполнены условия:

1) $h(\varnothing) = 0$;

2) $h(U) \le h(V)$, если $U \subseteq V \subseteq Q$;

3) $h(U \cup V) + h(U \cap V) \le h(U) + h(V)$ для любых подмножеств U и V множества Q;

4) если $h(U) = h(V)$ и $U \subseteq V \subseteq Q$, то $|S_U| = |S_V|$.

В частности, равенства $h(U) = 0$ и $|S_U| = 1$ равносильны. Если говорить неформально, то величина $h(U)$ (*энтропия* множества U) призвана задавать степень неопределённости, возникающую при попытках угадать зафиксированный элемент из S_U. Примером энтропийной функции на S может служить *комбинаторная энтропия*, определённая для всех U из $B(Q)$ как

$$h(U) = \log|S_U|,$$

(логарифмы берутся по фиксированному основанию $k > 1$), а также *энтропия Шеннона* (см. [2]), определённая для всех U из $B(Q)$ как

$$h(U) = -\Sigma P_U(x_U)\log P_U(x_U).$$

Во втором случае предполагается, что на множестве S определено распределение $P(x)$ (ненулевых) вероятностей случайной величины X, которое индуцирует распределение $P_U(x_U)$ случайной величины X_U на множестве S_U, а суммирование ведётся по всем x_U из S_U; при этом величину $h(U)$ принято называть *вероят-*

ностной энтропией, или *энтропией Шеннона*, случайной величины X_U. В обоих случаях все четыре свойства очевидны, известны или легко проверяются. В первом свойстве предполагается, что множество S_U при пустом U само не пусто, а состоит из единственного пустого набора (см. [2]). Вполне возможно, что имеют смысл и другие способы определения энтропийной функции, но перечисленные два являются основными. Имеет смысл при $U \subseteq V$ разность $h(V) - h(U)$ обозначать через $h(V|U)$ и называть *условной энтропией* множества V при известном U. Эта величина задаёт остаточную неопределённость вектора s_V при известном ограничении s_U (и известном S) и может принимать значения от 0 (когда s_V однозначно восстанавливается по известному s_U и S) до $h(V\setminus U)$. Отметим, что как комбинаторная, так и вероятностная энтропия h обладает ещё одним свойством:

5) если $h(U \cup V) + h(U \cap V) = h(U) + h(V)$, то $|S_{U\cup V}||S_{U\cap V}| = |S_U||S_V|$,

из которого следует, что при пустом пересечении $U \cap V$ равенство $h(U \cup V) = h(U) + h(V)$ влечёт $|S_{U\cup V}|=|S_U||S_V|$. Последнее означает, что компоненты s_U и s_V в наборе $s_{U\cup V}$ принимают все возможные значения из множеств S_U и S_V.

Совершенная срс

Пусть теперь $Q = P \cup \{D\}$, $D \notin P$ и h − энтропийная функция на непустом множестве $S \subseteq A^Q$. Набор (S, P, D, h) будем называть *совершенной срс*, если для каждого подмножества $U \subseteq P$ выполняется

$$h(U, D) = h(U) \text{ или } h(U, D) = h(U) + h(D).$$

(Здесь и далее из соображений краткости под знаком функции h опускаются знаки объединения и фигурные скобки; вместо них используются запятые.) В случае комбинаторной или вероятностной энтропии h будем говорить соответственно о *комбинаторной* или *вероятностной срс*. Те подмножества U, для которых выполняется первое условие, составляют сд $G(S, P)$, реализуемую данной срс. Как уже отмечалось, для таких подмножеств U имеется возможность для любого правила s из S определить значение s_D по проекции s_U. В это же время для неавторизованных в $G(S, P)$ подмножеств U выполняется второе условие, означающее, что участники этого множества при любом s из S, пытаясь угадать s_D по s_U, сталкиваются с максимально возможной неопределённостью. В случае комбинаторной срс это означает, что для любого набора y из S_U найдётся правило s в S такое, что $s_U = y$, обладающее любым наперёд заданным значением s_D из S_D. А в случае вероятностной срс это означает, что для любого правила s из S проекция s_U не несёт никакой информации о значении секрета s_D.

Скорость срс

В силу вышесказанного об энтропийной функции h возможность эффективного использования срс в значительной степени определяется величинами $h(p)$ для p из P. Чем они больше, тем больше требуется затрат для передачи, хранения и обработки долей. Вместе с тем без ущерба для разделения и восстановления секрета несущественным участникам можно не передавать их долей. В связи с этим эффективность совершенной срс (S, P, D, h), реализующей сд G, можно охарактеризовать величиной

$$r(S, P, D, h) = \min(h(p)/h(D)),$$

где min вычисляется по всевозможным существенным участникам p из $\cup G_0$. Эта величина называется *скоростью* срс (S, P, D, h). Имеет место

Теорема 1. Пусть совершенная срс (S, P, D, h) реализует сд G. Тогда для любого существенного в G участника p из P имеет место неравенство $h(p) \le h(D)$.

Доказательство. Так как участник p существенный в G, то для некоторого неавторизованного в G множества U множество $U \cup \{p\}$ авторизованно в G. Тогда $h(U) + h(p) \ge h(U, p) = h(U, p, D) \ge h(U, D) = h(U) + h(D)$. Отсюда $h(p) \ge h(D)$. Теорема доказана.

Из доказанной теоремы следует, что скорость определена корректно и принимает значения из отрезка [0, 1], если имеется хотя бы один существенный участник (то есть когда сд G не тривиальна − не пуста и отличается от $B(P)$). Если существенных участников нет, то скорость срс будем считать бесконечной и будем обозначать её знаком $+\infty$. Срс, имеющие скорость, равную 1 или $+\infty$, называются *идеальными*.

Одной из основных задач теории разделения секрета является задача создания (синтеза) срс для заданной сд. При этом требуется в некотором классе найти срс, реализующую заданную сд и имеющую по возможности максимальную скорость. Этим оправдывается введение для заданного класса C схем разделения секрета и заданной сд G на множестве P величины

$$r(C, G) = \sup r(S, P, D, h),$$

где sup берётся по всевозможным срс (S, P, D, h) из класса C, реализующим сд G. Данная величина носит название *скорости* сд G *в классе* C. Отметим, что скорость $r(C, G)$ данной сд G в классе C не обязана достигаться на некоторой срс из C, но может достигаться на некоторой последовательности срс. В частности, автор не исключает возможности существования сд, имеющей в некотором классе скорость, равную 1, но не имеющей в этом классе идеальной реализации.

Вычислительная эффективность срс

Введённая выше скорость схем разделения секрета в большей степени характеризует их коммуникационную (связанную с затратами при передаче и хранении долей секрета), нежели вычислительную эффективность. Последняя определяется вычислительными затратами, необходимыми для разделения и восстановления секрета. Эти затраты могут быть значительными вне зависимости от энтропии долей. Более прямо вычислительную эффективность срс (S, P, D, h) характеризует максимальная схемная сложность $l(S, P, D, h)$ функции

$$S_U \to S_D,$$

определённой для каждого авторизованного множества U следующим образом:

$$x \mapsto y, \text{ если } x = s_U \text{ и } y = s_D \text{ для некоторого } s \text{ из } S.$$

Естественно назвать эту величину *сложностью* срс (S, P, D, h). В связи с этим представляет интерес задача нахождения для заданной сд реализующей её срс, принадлежащей заданному классу и имеющей по возможности минимальную сложность. К сожалению, в известных автору работах данная задача не рассматривалась.

Срс, связанные с группами

Пусть $S \subseteq A^Q$, и для любого элемента q из Q пусть S_q – аддитивная (но не обязательно абелева) группа с множеством операторов K, действующих левым умножением [3, 4]. Через $\Pi_Q S_q$ обозначается прямое произведение этих групп, состоящее из всевозможных таких наборов s из A^Q, что $s_q \in S_q$ для любого q из Q, и являющееся группой с множеством операторов K, в которой

$$(s + w)_q = s_q + w_q, \ (cs)_q = cs_q$$

для любых s и w из $\Pi_Q S_q$ и любого c из K. Пусть также множество S является подгруппой группы $\Pi_Q S_q$ (и тогда подпрямым произведением групп S_q). В этой ситуации полезно иметь в виду, что для любого подмножества $U \subseteq Q$ и для нулевого набора 0 из $S_{Q\setminus U}$ каждое из множеств S_U и S^0 является (K-операторной) подгруппой группы $\Pi_U S_q$, а для любого набора y из $S_{Q\setminus U}$ множество S^y является смежным классом группы $S_{Q\setminus U}$ по подгруппе S^0. Пусть, наконец, $Q = P \cup \{D\}$ и для любого подмножества $U \subseteq P$ верно

$$|S_{U\cup\{D\}}| = |S_U| \text{ или } |S_{U\cup\{D\}}| = |S_U||S_{\{D\}}|. \qquad (1)$$

Тогда можно определить совершенную комбинаторную срс (S, P, D, h) с ранговой функцией h, такой, что $h(U) = log_{|S(D)|} S(U)$. Эта срс является одновременно и вероятностной для равномерного распределения вероятностей на S. В некоторых случаях условие (1) выполняется автоматически. Так происходит, например, когда группа S_D простая относительно множества операторов K, то есть имеет ровно две допустимые относительно K подгруппы (именно тривиальные подгруппы). В частности, именно этот случай имеет место, когда все S_q являются одномерными векторными пространствами над конечным полем K; в данном случае срс называется *векторной*, или *линейной одномерной*, или *срс Брикелла*. Если все S_q являются векторными пространствами над конечным полем K, но не обязательно одномерными, то срс называется *линейной*. Срс Брикелла представляют особый интерес в связи с их высокой эффективностью (они идеальны и обладают простыми алгоритмами разделения и восстановления секрета). В меньшей степени это относится к линейным срс (которые могут быть неидеальными). В ещё меньшей степени это относится к срс, связанным с группами. Последние не обязаны быть идеальными и для них не известны хорошие алгоритмы разделения и восстановления секрета.

Инвариантные классы сд

Введённые ранее операции взятия миноров над сд связаны с некоторыми операциями над срс. С этой целью рассмотрим энтропийную функцию h непустого подмножества $S \subseteq A^Q$ и для подмножества $U \subseteq Q$ определим функции

$$h_{Q\setminus U}: B(Q\setminus U) \to [0,+\infty), \ h^{(U)}: B(Q) \to [0,+\infty) \text{ и } h^{Q\setminus U}: B(Q\setminus U) \to [0,+\infty),$$

положив $h_{Q\setminus U}(V) = h(V)$ для $V \subseteq Q\setminus U$, $h^{(U)}(V) = h(U \cup V)$ для $V \subseteq Q$ и $h^{Q\setminus U} = h^{(U)}{}_{Q\setminus U}$. Несложно понять, что данные функции являются энтропийными для соответствующих множеств $S_{Q\setminus U}$, S^y и $S^y{}_{Q\setminus U}$, где набор y выбран произвольно из множества S_U. В действительности, имеет место

Теорема 2. Пусть (S, P, D, h) – совершенная срс, реализующая сд G, и $Q = P \cup \{D\}$. Тогда для любого подмножества $U \subseteq P$ и любого набора y из S_U верно:

1) набор $(S_{Q\setminus U}, P\setminus U, D, h_{Q\setminus U})$ является срс, реализующей сд $G_{P\setminus U}$;
2) набор $(S^y, P\setminus U, D, h^{(U)})$ является срс, реализующей сд $G^{(U)}$;
3) набор $(S^y{}_{Q\setminus U}, P\setminus U, D, h^{Q\setminus U})$ является срс, реализующей сд $G^{P\setminus U}$.

Следствие 1. Для любого r из $[0, 1] \cup \{+\infty\}$ класс C_r всех сд, чья скорость в классе всех срс не меньше r, инвариантен. В частности, инвариантен класс всех идеальных срс.

Отметим, что инвариантность того или иного класса сд позволяет охарактеризовать этот класс посредством запрещающего множества миноров, подобно тому, как это делается для наследственных классов графов [5]. Помимо класса C_r имеются и другие важные для приложений инвариантные классы сд, такие, как классы сд, имеющих векторную (или линейную) реализацию над заданным (или произвольным) конечным полем, класс групповых сд, класс совершенных вероятностных (или комбинаторных) сд, класс сд, имеющих реализацию порядка, равного заданной величине k или 1. Очевидно, при помощи операций пересечения и объединения из перечисленных классов снова получаются инвариантные классы. Отметим ещё одно

Следствие 2. Если существует реализующая сд G срс с энтропийной функцией h, то существует реализующая G срс с энтропийной функцией H, такой, что $H(p) \le h(p)$ для любого участника p сд G и $H(p) = 0$ для любого несущественного в G участника p.

Идеальный матроид

Для заданного класса C совершенных срс, представляющих практический интерес, имеет смысл задача конструктивного описания всех сд, имеющих реализацию в классе C. В частности, данная задача имеет смысл для класса всех идеальных совершенных срс. В настоящее время данная задача не решена, а наилучшие из известных условий связаны с понятием матроида. Напомним, что *полиматроидом* на множестве Q называется пара (Q, h), где h: $B(Q)\to[0, +\infty)$ – функция, обладающая свойствами (1) – (3) из определения энтропийной функции. В частности, энтропийная функция непустого подмножества $S \subseteq A^Q$ определяет полиматроид на множестве Q. Полиматроид (Q, h) называется *матроидом*, если дополнительно выполнено свойство

1) $h(U) \in \{0, ..., |U|\}$ для любого U из $B(Q)$.

При этом элементы множества Q называются *точками* матроида. Функция h называется *ранговой* функцией матроида. Её значение $h(U)$ называется *рангом* множества $U \in B(Q)$. Множества, чья мощность совпадает с рангом, называются *независимыми*, в отличие от остальных, *зависимых* множеств матроида. Минимальные по включению зависимые множества матроида называются его *циклами*. Матроид называется *связным*, если любые две его точки попадают в некоторый цикл. Известно, что матроид однозначно определяется всеми своими циклами, проходящими через любую его заданную точку. Имеет место следующая теорема

Теорема 3 (Брикелл, Дэвенпорт). Пусть (S, P, Q, H) – идеальная совершенная срс, реализующая сд G, не имеющую несущественных участников. Пусть также $H(D) = 1$. Тогда (Q, H) – матроид, циклами которого, проходящими через точку D, являются всевозможные множества $U \cup \{D\}$, $U \in G_0$.

Замечание. Требование $H(D) = 1$ не умаляет общности теоремы. Энтропийную функцию H совершенной срс с этим свойством всегда можно найти, разделив все значения произвольной энтропийной функции h на $h(D)$, то есть положив $H = h/(h(D))$.

Вариант доказательства данной теоремы можно найти в [6]. Без доказательства данная теорема приводится в [7]. В [6, 7] можно найти ссылки на оригинальную работу Дэвенпорта и Брикелла, содержащую доказательства данной теоремы. Ещё один вариант доказательства будет приведён далее. Для доказательства понадобятся вспомогательные леммы 1 и 2.

Лемма 1. Пусть в условиях теоремы 3 $V \in B(P)\backslash G$ и множество $U \in B(P)$ минимальное по включению, для которого выполняется условие $V \cup U \in G$. Тогда $H(U, V) = H(V) + |U|$. В частности, для базисного множества $U \in G_0$ верно $H(U) = |U|$.

***Доказательство леммы* 1.** Пусть множество U n-элементное, состоит из элементов $p_1, ..., p_n$. Тогда для любого m, $1 \le m \le n$, верно: $H(V, U, D) = H(V, U) \le H(V, U\setminus\{p_m\}) + H(p_m) = H(V, U\setminus\{p_m\}) + H(D) = H(V, U\setminus\{p_m\}, D) \le H(V, U, D)$. Очевидно, что здесь везде равенства. Учитывая это и используя свойства (2) и (3) из определения энтропийной функции, получаем $H(p_m) = H(V, U) - H(V, U\setminus\{p_m\}) \le H(V, p_1, ..., p_m) - H(V, p_1, ..., p_{m-1}) \le H(p_m) = H(D) = 1$. Очевидно, что и здесь равенства. Используя их при $m = 1, ..., n$, получаем, что $H(V, p_1) = H(V) + H(D) = H(V) + 1$ и $H(V, p_1, p_2) = H(V, p_1) + H(p_2) = H(V, p_1) + 1 = H(V) + 2$, ..., $h(V, U) = h(V) + |U|$. Что и требовалось доказать.

Лемма 2. Пусть в условиях теоремы 3 $p \in U$ и $U \in G_0$. Тогда для любого $V \in B(P)$, такого, что $U \subseteq V$, верно: $H(V\setminus\{p\}, D) = H(V, D) = H(V)$.

***Доказательство леммы* 2.** Легко понять, что $H(D) + H(U\setminus\{p\}) = H(p) + H(U\setminus\{p\}) \ge H(U) = H(U, D) \ge H(U\setminus\{p\}, D) = H(U\setminus\{p\}) + H(D)$. Отсюда следует, что $H(U\setminus\{p\}, D) = H(U, D) = H(U)$. В силу третьего свойства в определении энтропийной функции H данные равенства будут выполняться и после замены U на V. Лемма доказана.

***Доказательство теоремы* 3.** Прежде всего докажем, что (Q, H) – матроид. Для этого достаточно проверить целочисленность функции H. Более того, достаточно проверить целочисленность значений $H(U)$ лишь для авторизованных в G множеств U. Это следует из того, что не авторизованное множество V можно дополнить до авторизованного множества $U = V \cup W$ минимальным по включению множеством W, тогда по лемме 1 $H(U) = H(V, W) = H(V) + |W|$ и из целочисленности $H(U)$ следует целочисленность $H(V)$. Итак, будем

доказывать для авторизованного в G множества U, что значение $H(U)$ – целое. Назовём множество U m-множеством, если $U = U_1 \cup U_2$ для некоторого m-элементного множества U_1 и для некоторого не пересекающегося с ним базисного множества U_2. Ясно, что всякое авторизованное множество является m-множеством для некоторого натурального m. Индукцией по m покажем, что $H(U)$ – целое для m-множества U. Данное утверждение является следствием леммы 1 при $m = 0$, так как 0-множествами являются базисные множества. Пусть $M > 0$, и утверждение имеет место при всех $m < M$. Докажем его при $m = M$. Рассмотрим m-множество $U = U_1 \cup U_2$, где $|U_1| = m$ и U_2 из G_0. Предположим сначала, что U_2 – единственное базисное множество, включённое в U. Тогда по лемме 1 $H(U) = H(U_1) + |U_2|$ и нужно убедиться, что $H(U_1)$ – целое. Поскольку элементы множества U_1 существенны в P, найдётся базисное множество, имеющее непустое пересечение с U_1. Выберем некоторое такое базисное множество U_3 с минимальной по включению разностью $U_3 \backslash U_1$. Тогда множество $U_1 \cup U_3$ является m'-множеством для некоторого $m' < m$ (именно для $m' = |U_1 \backslash U_3|$) и по предположению $H(U_1, U_3)$ – целое. Так как в силу минимальности U_3 по лемме 1 $H(U_1, U_3) = H(U_1) + |U_3 \backslash U_1|$, то и значение $H(U_1)$ – целое. Возвращаясь назад, заключаем, что в рассмотренном случае $H(U)$ – целое. Осталось рассмотреть случай, когда U включает отличное от U_2 базисное множество, содержащее некоторый элемент p из U_1. Так как множество $U \backslash \{p\}$ авторизовано (ибо оно включает U_2), то $H(U \backslash \{p\}, D) = H(U)$. По лемме 2 заключаем, что $H(U \backslash \{p\}) = H(U)$. Поскольку $U \backslash \{p\} = (U_1 \backslash \{p\}) \cup U_2$, множество $U \backslash \{p\}$ является $(m-1)$-множеством. По предположению индукции значение $H(U) = H(U \backslash \{p\})$ – целое. Итак, доказано, что пара (Q, H) является матроидом. Обратим своё внимание на циклы данного матроида. Прежде всего отметим, что циклами матроида являются такие множества точек, чей ранг на 1 меньше мощности и чьё любое собственное подмножество имеет ранг, совпадающий с мощностью. В силу этого для любого цикла $U \cup \{D\}$, где $D \notin U$, верно, что $|U| = H(U, D) = H(U) = H(U \backslash \{p\}, D) = H(U \backslash \{p\}) + 1$ для любого p из U. Это означает, что U – минимальное авторизованное в G, то есть принадлежит G_0. Обратно, если U из G_0, то, учитывая леммы 1 и 2, имеем $|U| = H(U) = H(U, D) = H(U \backslash \{p\}, D)$ для всех p из U. Следовательно, $U \cup \{D\}$ – цикл. Теорема доказана.

Итак, теорема 3 показывает, что энтропийная функция совершенной идеальной срс является ранговой функцией матроида. Матроиды, получающиеся таким путём, называются *идеальными*. В настоящее время известны неидеальные матроиды [7]. В [7] указывается, что матроид, отвечающий совершенной комбинаторной срс, связный.

Срс и сд Брикелла

Рассмотрим срс Брикелла над конечным полем K и реализуемые ими сд. Такие сд называются *сд Брикелла* над полем K. Установим необходимые и достаточные условия того, что заданная сд является сд Брикелла над заданным конечным полем. Итак, пусть (S, P, Q, h) – срс Брикелла над конечным полем K, реализующая сд G. Выберем в векторном пространстве S порождающий его набор R векторов $r_1, ..., r_m$, линейная оболочка $<r_1, ..., r_m>$ которых совпадает с S. Для любого q из Q через $R(q)$ обозначим набор $(r_{1q}, ..., r_{mq})$ и для любого U из $B(Q)$ через $R(U)$ обозначим множество наборов $R(q)$ для всевозможных q из U. Возникшее таким образом отображение

$$R: Q \to K^m,\ R: q \mapsto R(q)$$

называется *реализацией Брикелла* срс (S, P, Q, h) и сд G. Отметим пару свойств реализации R, имеющих место для любого подмножества U из $B(P)$.

Свойство 1. Множество U тогда и только тогда авторизовано в G, когда вектор $R(D)$ принадлежит линейной оболочке $<R(U)>$. Это следует из того, что $\operatorname{rank} S_V = \operatorname{rank} <R(V)>$ для любого V из $B(Q)$, что в свою очередь, является простой переформулировкой следующего известного свойства из [3]: строчный ранг матрицы над полем совпадает с её столбцовым рангом.

Свойство 2. Множество U тогда и только тогда не авторизовано в G, когда существует вектор $v(U)$ в K^m, такой, что $v(U) \bullet R(D) = 1$ (здесь и далее точкой обозначаем скалярное произведение) и $v(U) \bullet B(p) = 0$ для любого p из U. Действительно, в силу предыдущего свойства $U \in G \Leftrightarrow <R(D)> \subseteq <R(U)> \Leftrightarrow <R(U)>\perp \subseteq <R(D)>\perp$ (в последнем выражении речь идёт об ортогональном дополнении). Следовательно, условие $U \in G$ равносильно существованию вектора $v(U)$ в $<R(U)>\perp \backslash <R(D)>\perp$. Иными словами, вектор $v(U)$ ортогонален всем векторам $R(p)$, таким, что $p \in U$, и не ортогонален вектору $B(D)$. Очевидно, вектор $v(U)$ можно выбрать так, что $v(U) \bullet R(D) = 1$. Тем самым свойство 2 доказано. Имеет место

Теорема 4. Пусть G – сд с множеством участников P. Если G – сд Брикелла над конечным полем K, то существует пара функций

$$d: G_0 \times P \to K \text{ и } d^*: G^*_0 \times P \to K,$$

обладающих следующими свойствами:

1) $d(U, p) \neq 0$ и $d^*(V, q) \neq 0$, если $p \in U \in G_0$ и $q \in V \in G^*_0$;

2) $d(U, p) = 0$ и $d^*(V, q) = 0$, если $p \notin U \in G_0$ и $q \notin V \in G^*_0$;

3) $\Sigma d(U, p) d^*(V, p) = 1$ для любых $U \in G_0$ и $V \in G^*_0$, где суммирование ведётся в K по всевозможным p из P.

Обратно, если существуют функции d и d^*, обладающие свойствами 2) и 3), то сд G является сд Брикелла над K.

Доказательство. Необходимость. Пусть G – сд Брикелла над конечным полем K с реализацией R: $Q \to K^m$. Пусть V – множество из G^*_0. Тогда $P \setminus V$ – максимальное неавторизованное в G. В силу второго свойства существует вектор $v = v(P \setminus V)$ в K^m, такой, что $v \bullet R(D) = 1$ и $v \bullet R(p) = 0$ для любого $p \in P \setminus V$. Отметим, что в силу максимальности множества $P \setminus V$ скалярные произведения $v \bullet R(p)$ при $p \in V$ не равны нулю. Пусть также U – минимальное авторизованное в G множество из G_0. Тогда в силу первого свойства вектор $R(D)$ выражается линейно через вектора $B(p)$, где p из U, с ненулевыми (в силу минимальности U) коэффициентами. Обозначим эти коэффициенты через $d(U, p)$ так, что $B(D) = \Sigma d(U, p) R(p)$, и положим $d(U, p) = 0$ для всех p из $P \setminus U$. Тогда $1 = v \bullet R(D) = v \bullet \Sigma d(U, p) R(p) = \Sigma d(U, p)(v \bullet R(p))$. Остаётся положить $d^*(V, p) = v \bullet R(p)$. Докажем достаточность. С этой целью будем считать заданными величины $d(U, p)$ и $d^*(V, p)$, обладающие вторым и третьим свойствами. Пусть множество G^*_0 состоит из множеств $V_1, ..., V_m$. Зададим функцию R: $Q \to K^m$ так, что $R(D) = (1, ..., 1)$ и $R(p) = (d^*(V_1, p), ..., d^*(V_m, p))$ для всех p из P. Легко понять, что для любого U из G_0 вектор $R(D)$ выражается линейно через вектора $R(p)$, p из U с коэффициентами $d(U, p)$; это следует из второго и третьего свойств. Если же U – максимальное не авторизованное в G множество, то разность $P \setminus U$ совпадает с некоторым множеством V_i. Тогда всякий вектор $R(p)$, где p из U, имеет 0 в i-й координате, а значит, вектор $R(D) = (1, ..., 1)$ нельзя выразить линейной комбинацией этих векторов. Отмеченные свойства функции R говорят о том, что она является реализацией Брикелла сд G. Теорема доказана. Непосредственно из теоремы получаем

Следствие 3. Сд G на множестве P тогда и только тогда является сд Брикелла над конечным полем K, когда над этим полем относительно переменных $d(U, p)$ и $d^*(V, q)$, где $p \in U \in G_0$ и $q \in V \in G^*_0$, разрешима система уравнений $\Sigma d(U, p) d^*(V, p) = 1$ (сумма по всем p из $U \cap V$) для всевозможных значений $U \in G_0$ и $V \in G^*_0$.

В случае двухэлементного поля $K = Z_2$ получаем

Следствие 4 [10]. Сд G тогда и только тогда является сд Брикелла над полем Z_2, когда для любого множества U из G и любого множества V из G^*_0 пересечение $U \cap V$ содержит нечётное число элементов.

Данное следствие было опубликовано с доказательством в [6]. Теорема 4 является непосредственным обобщением этого результата.

Замечание. Следствие 3 сводит вопрос о существовании реализации Брикелла над полем K для заданной сд G к разрешимости над K некоторой системы уравнений. Причём, как видно из доказательства теоремы 4, решение данной системы даёт реализацию Брикелла данной сд, впрочем, возможно неэффективную (с большим m). В [9] описывается алгоритм, распознающий совместность системы полиномиальных уравнений над алгебраически замкнутым полем. В соответствии с этим алгоритмом нужно найти редуцированный базис Грёбнера идеала, порождённого системой многочленов (определяющих систему уравнений). Тогда система оказывается совместной, если редуцированный базис не состоит из одного единичного многочлена. При нахождении редуцированного базиса методом Бухбергера из [9] все вычисления происходят в кольце многочленов над фиксированным полем. В частности, если коэффициенты системы целочисленные (как в нашем случае), то все вычисления будут происходить в простом поле заданной характеристики. Это приводит ещё к одному интересному следствию.

Следствие 5. Для любого простого p существует алгоритм, распознающий для произвольной заданной сд свойство иметь реализацию Брикелла над конечным полем характеристики p.

Замечание. Отметим ещё одно свойство системы из следствия 3. Оказывается, переменные $d(U, p)$ можно выразить через переменные $d^*(V, q)$, поскольку для любого U из G_0 и p из U в G^*_0 найдётся такое V, что в системе имеется уравнение $d(U, p) d^*(V, p) = 1$. Аналогично, переменные $d^*(V, q)$ выражаются через переменные $d(U, q)$. Указанные возможности следуют из леммы 3.

Лемма 3. Для любой сд G, любого множества U из G_0 и любого p из U найдётся такое множество V в G^*_0, что $U \cap V = \{p\}$.

Доказательство. Обязательно найдётся максимальное по включению неавторизованное в G множество W, такое, что $U \setminus \{p\} \subseteq W \subseteq P \setminus \{p\}$. Так как дополнение $V = P \setminus W$ принадлежит G^*_0 и $U \cap V = \{p\}$, лемма доказана.

Нижние оценки порядка реализации

Предположим, что сд G является сд Брикелла над некоторым конечным полем K, содержащим q элементов. Представляется интересной задача нахождения наименьшего такого значения q для заданной сд G. Также имеет смысл задача нахождения нижней оценки для q. При решении данных задач может оказаться полезной следующая лемма. Далее будет показано, как этой леммой следует пользоваться.

Лемма 4. Пусть G – сд Брикелла над полем порядка q на множестве участников $P = A \cup \{a_1, ..., a_r\}$, где элементы $a_1, ..., a_r$ все различные из $P \setminus A$ и для любого i из $\{1, ..., r\}$ существует множество A_i в G_0, такое, что $a_i \in A_i$ и $A_i \subseteq A \cup \{a_1, ..., a_i\}$. Тогда для любого элемента a_0 из A число таких множеств B в G^*_0, что пересечение $B \cap (A \setminus \{a_0\})$ пусто, не превосходит q.

Доказательство. В системе уравнений из теоремы 4 в этом случае для всякого множества B из G^*_0 будут присутствовать уравнения:

$$d(A_1, a_0)d^*(B, a_0) + d(A_1, a_1)d^*(B, a_1) = 1,\ \ d(A_2, a_0)d^*(B, a_0) + d(A_2, a_1)d^*(B, a_1) + d(A_2, a_2)d^*(B, a_2) = 1, ...$$

Отсюда видно, что значение $d^*(B, a_0)$ однозначно определяет все значения $d^*(B, a)$ для всевозможных a из P и таким образом однозначно определяет само множество B. Следовательно, имеется не более q возможностей для выбора B. Лемма доказана.

Пороговая сд

Рассмотрим пример использования леммы 4.

Структура доступа G на n-элементном множестве P называется (n, k)-*пороговой*, если её базис, множество G_0, состоит из всевозможных k-элементных подмножеств множества P. Хорошо известно, что для любых целых k и n, таких, что $1 \le k \le n$, (n, k)-пороговая сд является сд Брикелла над некоторым конечным полем. Об этом можно прочитать, например, в [7, 10]. Несложно понять, что для заданных целого положительного k и целого положительного q, являющегося степенью простого числа, существует наибольшее n, при котором существует сд Брикелла над q-элементным полем, реализующая (n, k)-пороговую сд. Для такого наибольшего n положим $m(k, q) = n + 1$. Несложно понять, что $m(k, q)$ – это наибольшее число векторов в векторном пространстве над q-элементным полем, из которых любые k линейно независимы и любые $k + 1$ линейно зависимы. В некоторых случаях значение $m(k, q)$ найти несложно. Так, величина $m(1, q)$ совпадает с длиной кода Хэмминга над q-элементным полем и, следовательно, равняется $q + 1$. Определение точного вида функции $m(k, q)$ в общем случае является давней комбинаторной задачей, не решённой и по сей день [7, 11]. Предполагается, что $m(k, q) = q + 1$ за исключением случая $m(3, q) = (q - 1, q) = q + 2$ при чётном q. Следующая теорема даёт нижнюю оценку порядка реализации Брикелла для (n, k)-пороговой сд и одновременно даёт верхнюю оценку $m(k, q) \le q + k - 1$ для величины $m(k, q)$, лишь на 1 худшую, чем оценка теоремы 11 из [11].

Теорема 5. Если над q-элементным полем существует реализация Брикелла (n, k)-пороговой сд и $k \ge 2$, то $n \le q + k - 2$.

Доказательство данной теоремы опирается на лемму 4. Пусть G – (n, k)-пороговая сд на n-элементном множестве P. Тогда G_0 состоит из всевозможных k-элементных подмножеств множества P и G^*_0 состоит из всевозможных $(n - k + 1)$-элементных подмножеств. Это позволяет в условиях леммы 4 выбрать $(k–1)$-элементное множество A. Тогда, с одной стороны, число подмножеств B, о которых идёт речь в лемме, равно $n - k + 2$. С другой стороны, это число по лемме 4 не превосходит q. Таким образом получаем неравенство $n - k + 2 \le q$. Теорема доказана.

ЛИТЕРАТУРА

1. *Яблонский С.В.* О классах функций алгебры логики, допускающих простую схемную реализацию // УМН. 1957. Т. XII. Вып. 6 (78). С. 189 – 196.
2. *Алон Н.*, *Спенсер Д.Ж.* Вероятностный метод. М.: Бином. Лаборатория знаний, 2007. 320 с.
3. *Ван дер Варден Б.Л.* Алгебра. Определения, теоремы, формулы. 3-е изд., стер. – СПб.: Лань, 2004. 624 с.
4. *Каргаполов М.И.*, *Мерзляков Ю.И.* Основы теории групп. М.: Наука, 1972. 240 с.
5. *Дистель Р.* Теория графов. Новосибирск: Изд-во Ин-та математики, 2002. 336 с.
6. *Marti-Ferre J.*, *Padro C.* On secret sharing schemes, matroids and polymatroids // Cryptology ePrint Archive, Report 2006/077, http://eprint.iacr.org/2006/077.
7. *Введение* в криптографию / Под общ. ред. В.В. Ященко. 3-е изд., доп. М.: МЦНМО «ЧеРо», 2000. 288 с.
8. *Marti-Ferre J.*, *Padro C.* Secret sharing schemes on sparse homogeneous access structures with rank three // The electronic journal of combinatorics 11 (2004), Research papper 72, 16 p. (electronic).
9. *Кокс Д.*, *Литтл Дж.*, *О'Ши Д.* Идеалы, многообразия и алгоритмы. Введение в вычислительные аспекты алгебраической геометрии и коммутативной алгебры. М.: Мир, 2000. 687 с.
10. *Агибалов Г.П.* Избранные теоремы начального курса криптографии. Томск: Изд-во НТЛ, 2005. 116 с.
11. *Мак-Вильямс Ф. Дж.*, *Слоэн Н. Дж. А.* Теория кодов, исправляющих ошибки. М.: Связь, 1979. 744 с.

2008 Математические методы криптографии № 2(2)

DOI 10.17223/20710410/2/12

УДК 519.7

СВОЙСТВА НЕКОТОРЫХ АЛГОРИТМОВ ШИФРОВАНИЯ ФЕЙСТЕЛЯ ОТНОСИТЕЛЬНО ДВУХ ГРУПП СПЛЕТЕНИЯ[1]

М.А. Пудовкина

Московский инженерно-физический институт (*государственный университет*)

E-mail: maricap@rambler.ru

Одной из наиболее часто встречающихся конструкций, применяемой при синтезе блочных алгоритмов шифрования, является схема Фейстеля. В работе исследуются свойства некоторых алгоритмов шифрования на основе схемы Фейстеля относительно двух групп сплетений. Описаны слабости такого класса алгоритмов шифрования.

Ключевые слова: *алгоритм шифрования Фейстеля*, *сплетения групп подстановок*, *импримитивная группа.*

Одной из наиболее часто встречающихся конструкций, применяемых при синтезе блочных алгоритмов шифрования, является схема Фейстеля. Напомним (см., например, [1]), что схема Фейстеля задается преобразованием $g_{\pi_k} : V_m \times V_m \to V_m \times V_m$, $g_{\pi_k} : (\alpha,\beta) \to (\beta, \beta^{\pi_k} \oplus \alpha)$, где V_m – множество всех m-мерных двоичных векторов над полем $GF(2)$, $\pi_k: V_m \to V_m$, $k \in V_m$, $\oplus$ – операция векторного сложения в V_m. Преобразование π_k зависит от ключа k. Алгоритмы шифрования на основе схемы Фейстеля будем называть алгоритмами шифрования Фейстеля.

Групповые свойства алгоритмов шифрования Фейстеля приведены, например, в работах [2 – 4]. Так, в [4] рассматривался случай, когда подстановка π_k принадлежит импримитивной группе, и строилась система блоков импримитивности для алгоритма шифрования DES с измененными s-боксами.

Две импримитивные группы $S_2 \wr S_{2^{n-1}}$ и $S_{2^{n-1}} \wr S_2$ большого порядка, которые максимально близки к примитивным группам, неоднократно возникали в различных смежных областях, например в работе [5] при классификации групп автоморфизмов графов орбиталов подсхем схемы Хемминга и в работе [6]. Поэтому представляет интерес рассмотреть свойства алгоритмов шифрования Фейстеля с подстановкой π_k, принадлежащей двум этим группам.

В данной работе исследуются свойства алгоритмов шифрования Фейстеля таких, что $\pi \in \{S_2 \wr S_{2^{m-1}}, S_{2^{m-1}} \wr S_2\}$, а $\beta^{\pi_k} \in \{(\beta \oplus k)^{\pi}, \beta^{\pi} \oplus k\}$ для всех k, $\beta \in V_m$. Описаны слабости такого класса алгоритмов шифрования. Получено, что если $\pi \in S_{2^{m-1}} \wr S_2$, то по шифртексту возможно получить некоторую информацию об открытом тексте без знания ключа. Если же $\pi \in S_2 \wr S_{2^{m-1}}$, то на множестве ключей можно ввести отношение эквивалентности, и задача определения ключа сводится к проблеме нахождения какого-нибудь представителя из класса эквивалентности, которому принадлежит ключ.

Всюду ниже придерживаемся следующих обозначений: $S(X)$ – симметрическая группа подстановок на множестве X, N – множество натуральных чисел; $m \in N$; $n = 2m$; $\overline{a,b} = a, a+1, ..., b$, $a < b$; $\Phi_n = \{f_\pi \in S(V_m \times V_m) \mid f_\pi : (\alpha,\beta) \to (\beta, \beta^{\pi} \oplus \alpha)\}$ – множество алгоритмов шифрования Фейстеля; e – тождественная подстановка, $g^s = s^{-1}gs$ для подстановок g, $s \in S(X)$.

Будем отождествлять 2-адическое представление элемента a с представлением его в виде двоичного вектора.

1. Свойства алгоритмов шифрования Фейстеля относительно группы $S_2 \wr S_{2^{m-1}}$

Если подстановка $\pi \in S(V_m)$ фиксирована, $f_{\pi_k} \in \Phi_n$ и $\alpha^{\pi_k} \in \{(\alpha \oplus k)^{\pi}, \alpha^{\pi} \oplus k\}$ для любого $\alpha \in V_m$, то будем использовать обозначение $f_k = f_{\pi_k}$.

Рассмотрим двоичную последовательность $\overline{a} = a_1 a_2, ...,$

$$a_i = \begin{cases} 1, & \text{если } i \equiv 1, 2 \pmod 3, \\ 0, & \text{если } i \equiv 0 \pmod 3, \end{cases}$$

являющуюся решением рекуррентного соотношения $a_i = a_{i-1} \oplus a_{i-2}$ в $GF(2)$, $a_0 = 0$, $a_1 = 1$, $i = 2, 3, \ldots$

[1] Работа выполнена при поддержке гранта Президента РФ НШ №4.2008.10.

Утверждение 1. Пусть $\pi \in S_2 \wr S_{2^{m-1}}$, $f_k \in \Phi_n$, $\alpha^{\pi_k} \in \{(\alpha \oplus k)^{\pi}, \alpha^{\pi} \oplus k\}$ для любого $k \in V_m$. Тогда

1) $\pi_k \in S_2 \wr S_{2^{m-1}}$ для любого $k \in V_m$;

2) $$(\alpha,\beta)^{\prod_{i=1}^{l} f_{k_i} \oplus \theta_i} = \left(\alpha^{(l)} \oplus \sum_{i=1}^{l-1} a_{l-i}\theta_i, \beta^{(l)} \oplus \sum_{i=1}^{l} a_{l-i+1}\theta_i\right), \quad (1)$$

где $l \in N$, $(\alpha,\beta)^{\prod_{i=1}^{l} f_{k_i}} = (\alpha^{(l)}, \beta^{(l)})$ и $\theta_i \in \{\vec{0}, \vec{1}\}$, $i = \overline{1,l}$.

Доказательство. 1) Включение $s \in S_2 \wr S_{2^{m-1}}$ выполняется тогда и только тогда, когда $(\alpha \oplus \vec{1})^s = \alpha^s \oplus \vec{1}$ для любого $\alpha \in V_m$. Очевидно, что $(\alpha \oplus \vec{1})^{\pi} \oplus k = (\alpha \oplus k)^{\pi} \oplus \vec{1}$ для любого $k \in V_m$. Следовательно, $\pi_k \in S_2 \wr S_{2^{m-1}}$.

2) Пусть $\theta_i \in \{\vec{0}, \vec{1}\}$, $i = 1, 2, \ldots$ Рассмотрим последовательность $y_i = y_{i-1} \oplus y_{i-2} \oplus \theta_i$, $y_0 = y_{-1} = 0$, $i = 1, 2, \ldots$ Методом математической индукции нетрудно показать, что $y_i = \sum_{j=1}^{i} a_{i-j+1}\theta_j$.

Равенство

$$(\alpha,\beta)^{\prod_{i=1}^{l} f_{k_i} \oplus \theta_i} = \left(\alpha^{(l)} \oplus y_{l-1}, \beta^{(l)} \oplus y_l\right)$$

доказывается применением метода математической индукции с учетом следующих соотношений:

$$(\alpha,\beta)^{f_k \oplus \theta_1} = \left(\beta, \beta^{\pi_{k \oplus \theta_1}} \oplus \alpha\right) = \begin{cases} (\beta, ((k \oplus \theta_1) \oplus \beta)^{\pi} \oplus \alpha) \\ (\beta, (\beta^{\pi} \oplus (k \oplus \theta_1)) \oplus \alpha) \end{cases} = (\beta, \beta^{\pi_k} \oplus \alpha \oplus \theta_1) =$$

$$= (\alpha^{(1)}, \beta^{(1)} \oplus \theta_1) = (\alpha^{(1)} \oplus y_0, \beta^{(1)} \oplus y_1) = (\alpha \oplus \theta_1, \beta)^{f_k},$$

$$(\alpha,\beta)^{\prod_{i=1}^{2} f_{k_i} \oplus \theta_i} = (\beta^{\pi_{k_1}} \oplus \alpha \oplus \theta_1, (\beta^{\pi_{k_1}} \oplus \alpha)^{\pi_{k_2}} \oplus \beta \oplus \theta_1 \oplus \theta_2) = (\alpha^{(2)} \oplus y_1, \beta^{(2)} \oplus y_2).$$

Утверждение доказано.

Ключи, принадлежащие одному блоку импримитивности $A_k = \{k, k \oplus \vec{1}\}$, $k \in \overline{\{0, 2^{m-1} - 1\}}$, группы $S_2 \wr S_{2^{m-1}}$, будем называть 1-эквивалентными; также ключи $k^{(j)} = (k_1^{(j)}, \ldots, k_l^{(j)}) \in V_m{}^l$, $j = 1, 2$, будем называть 1-эквивалентными, если $k_i^{(1)}, k_i^{(2)}$ – 1-эквивалентны для любого $i \in \overline{\{1,l\}}$. Очевидно, что отношение 1-эквивалентности на множестве $V_m{}^l$ является отношением эквивалентности.

Предположим, что для l-раундового алгоритма шифрования Фейстеля с раундовыми функциями $f_{\pi_{k_i}}$, $i = \overline{1,l}$, удовлетворяющими условиям утверждения 1, ключи $k_1, \ldots, k_l$ выбираются случайно независимо и равновероятно из V_m. Опишем эквивалентные ключи такого алгоритма, где ключи считаются эквивалентными в смысле стандартного определения (см., например, [7]).

Следствие 1. Пусть выполнены условия утверждения 1, $l \geq 2$, $k = (k_1, \ldots, k_l) \in V_m{}^l$. Пусть также $\theta = (\theta_1, \ldots, \theta_l)$, $\theta' = (\theta'_1, \ldots, \theta'_l)$ – произвольные векторы из $\{\vec{0}, \vec{1}\}^l$. Тогда ключи $k + \theta$, $k + \theta'$ эквивалентны, если выполняются равенства: $\sum_{i=1}^{l-1} a_{l-i}(\theta_i \oplus \theta'_i) = \vec{0}$ и $\theta_l = \theta'_l$. Число эквивалентных ключей, являющихся также 1-эквивалентными, равно 2^{l-2}.

Доказательство. Условия $\sum_{i=1}^{l-1} a_{l-i}(\theta_i \oplus \theta'_i) = \vec{0}$ и $\theta_l = \theta'_l$ непосредственно следуют из равенства (1). Подсчитаем число эквивалентных ключей, являющихся также 1-эквивалентными, равное числу решений уравнения $\sum_{i=1}^{l-1} a_{l-i}\tilde{\theta}_i = \vec{0}$, и $\tilde{\theta}_l = \vec{0}$, где $\tilde{\theta}_i = \theta_i \oplus \theta'_i \in \{\vec{0}, \vec{1}\}$. Поскольку число нулей в последовательности $a_1, \ldots, a_{l-1}$ равно $\left\lfloor \frac{l-1}{3} \right\rfloor$, а число единиц – $l - 1 - \left\lfloor \frac{l-1}{3} \right\rfloor$, то число решений уравнения $\sum_{i=1}^{l-1} a_{l-i}\tilde{\theta}_i = \vec{0}$ равно $2^{\left\lfloor \frac{l-1}{3} \right\rfloor} \cdot 2^{l-1-\left\lfloor \frac{l-1}{3} \right\rfloor - 1} = 2^{l-2}$. Следствие доказано.

Следующий алгоритм при фиксированном открытом тексте $(\alpha_1, \beta_1), \ldots, (\alpha_t, \beta_t) \in V_{2m}$ и соответствующем шифртексте $(\alpha_1^{(l)},\beta_1^{(l)}),...,(\alpha_t^{(l)},\beta_t^{(l)})$, $l \geq 1$, полученном на неизвестном ключе $k = (k_1,...,k_l) \in {V_m}^l$, позволяет уменьшить трудоемкость определения ключа k по сравнению с полным перебором.

Пусть $i \geq 1$. На i-м этапе опробования по схеме выбора без возвращения из множества всех блоков импримитивности $A = \{A(k),\ k = \overline{0, 2^{m-1}}\}$ упорядоченно выбираем l блоков, $r_1^{(i)}, ..., r_l^{(i)}$. Полагаем $k_1^{(i)} = k_{r_1^{(i)}},...,k_l^{(i)} = k_{r_l^{(i)}}$, где $k_j^{(i)}$ – j-й опробуемый ключ на i-м этапе, $k_{r_j^{(i)}}$ – произвольный элемент из $A(r_j^{(i)})$, $j = \overline{1,l}$. Если $(\alpha_j,\beta_j)^{f_{k_1^{(i)}...k_l^{(i)}}} = (\alpha_j^{(l,i)},\beta_j^{(l,i)}) \in \left\{(\alpha_j^{(l)},\beta_j^{(l)}),(\alpha_j^{(l)},\beta_j^{(l)}) + \vec{1}\right\}$, где $f_{k_1^{(i)}...k_l^{(i)}} = \prod_{j=1}^{i} f_{k_j^{(i)}}$, для любого $j \in \{\overline{1,l}\}$, то найден представитель $k^{(i)} = (k_1^{(i)},...,k_l^{(i)})$ класса 1-эквивалентности, содержащего истинный ключ. В противном случае переходим к этапу i+1.

Из следствия 1 следует способ нахождения ключа из класса эквивалентности, содержащего истинный ключ, по данному представителю $k^{(i)} = (k_1^{(i)},...,k_l^{(i)})$ класса 1-эквивалентности, которому принадлежит k. Для этого

- если $\beta_j^{(l)} = \beta_j^{(l,i)}$, то полагаем $k_l = k_l^{(i)}$, иначе $k_l = k_l^{(i)} + \vec{1}$;
- если $\alpha_j^{(l)} = \alpha_j^{(l,i)}$, то полагаем $k_{l-1} = k_{l-1}^{(i)}$, иначе $k_{l-1} = k_{l-1}^{(i)} + \vec{1}$;
- полагаем $k_j = k_j^{(i)}$, $j = \overline{1,l-2}$.

Трудоемкость предложенного алгоритма равна $2^{l(m-1)}$ э.о. и в 2^l раз меньше трудоемкости полного перебора, равной 2^{lm} э.о., где э.о. – элементарная операция.

2. Свойства алгоритмов шифрования Фейстеля относительно группы $S_{2^n-1} \int S_2$

Пусть $(\alpha,\beta)^{f_k} = (\beta, \alpha \oplus \beta^{\pi_k})$. Для векторов $(\alpha, \beta) \in V_m$, удовлетворяющих сравнению $||\alpha|| \equiv ||\beta|| \pmod 2$, будем использовать обозначение $\alpha \sim_2 \beta$. Также обозначим

$$(\alpha,\beta) \xrightarrow{f} (\alpha',\beta'),$$

если $(\alpha,\beta)^f = (\alpha^{(l)},\beta^{(l)})$, $f \in \Phi_n$, и $\alpha^{(l)} \sim_2 \alpha'$, $\beta^{(l)} \sim_2 \beta'$.

Утверждение 2. Пусть $\pi \in S_{2^m-1} \int S_2$, $f_k \in \Phi_n$, $\alpha^{\pi_k} \in \{(\alpha \oplus k)^\pi, \alpha^\pi \oplus k\}$ для любого $k \in V_m$. Тогда для любого натурального числа l и любых $k_1, \ldots, k_l \in V_m$ справедливо включение:

1. $\left(\prod_{j=1}^{l} f_{k_j}\right)^3 \in S_{2^n-1} \int S_2$, если $l \not\equiv 0 \pmod 3$.
2. $\left(\prod_{j=1}^{l} f_{k_j}\right)^2 \in S_{2^n-1} \int S_2$, если $l \equiv 0 \pmod 3$.

Доказательство. Методом математической индукции нетрудно показать, что для любого $i \in N$ справедливы равенства:

$$(\alpha,\beta) \xrightarrow{f_{k_1}...f_{k_{3\cdot i}}} (\alpha \oplus w_1^{(3i)}(k_1,...,k_{3i}), \beta \oplus w_2^{(3i)}(k_1,...,k_{3i})),$$

$$(\alpha,\beta) \xrightarrow{f_{k_1}...f_{k_{3\cdot i+1}}} (\beta \oplus w_1^{(3i+1)}(k_1,...,k_{3i+1}), \alpha \oplus \beta \oplus w_2^{(3i+1)}(k_1,...,k_{3i+1})),$$

$$(\alpha,\beta) \xrightarrow{f_{k_1}...f_{k_{3\cdot i+2}}} (\alpha \oplus \beta \oplus w_1^{(3i+2)}(k_1,...,k_{3i+2}), \alpha \oplus w_2^{(3i+2)}(k_1,...,k_{3i+2})),$$

где $w_j^{(3i+v)} : V_m^{3i+v} \to \{0,1\}$ – некоторые аффинные функции, $v \in \{0, 1, 2\}$, $j \in \{1, 2\}$.

Рассмотрим три случая. Если $l \equiv 0 \pmod 3$, то

$$(\alpha,\beta) \xrightarrow{f_{k_1}...f_{k_l}} (\alpha \oplus w_1^{(l)}(k_1,...,k_l), \beta \oplus w_2^{(l)}(k_1,...,k_l)) \xrightarrow{f_{k_1}...f_{k_l}}$$
$$(\alpha \oplus w_1^{(l)}(k_1,...,k_l) \oplus w_1^{(l)}(k_1,...,k_l), \beta \oplus w_2^{(l)}(k_1,...,k_l) \oplus w_2^{(l)}(k_1,...,k_l)) \sim_2 (\alpha,\beta).$$

Если $l \equiv 1 \pmod 3$, то

$$(\alpha,\beta) \xrightarrow{f_{k_1}\dots f_{k_l}} (\beta \oplus w_1^{(l)}(k_1,\dots,k_l), \alpha \oplus \beta \oplus w_2^{(l)}(k_1,\dots,k_l)) \xrightarrow{f_{k_1}\dots f_{k_l}} (\alpha \oplus \beta \oplus w_2^{(l)}(k_1,\dots,k_l) \oplus w_1^{(l)}(k_1,\dots,k_l),$$

$$w_1^{(l)}(k_1,\dots,k_l) \oplus \alpha) \xrightarrow{f_{k_1}\dots f_{k_l}} (w_1^{(l)}(k_1,\dots,k_l) \oplus w_1^{(l)}(k_1,\dots,k_l) \oplus \alpha, \alpha \oplus \beta \oplus w_2^{(l)}(k_1,\dots,k_l) \oplus w_1^{(l)}(k_1,\dots,k_l) \oplus$$

$$\oplus w_1^{(l)}(k_1,\dots,k_l) \oplus \alpha \oplus w_2^{(l)}(k_1,\dots,k_l)) \sim_2 (\alpha,\beta).$$

Если $l \equiv 2 \pmod 3$, то

$$(\alpha,\beta) \xrightarrow{f_{k_1}\dots f_{k_l}} (\alpha \oplus \beta \oplus w_1^{(l)}(k_1,\dots,k_l), \alpha \oplus w_2^{(l)}(k_1,\dots,k_l)) \xrightarrow{f_{k_1}\dots f_{k_l}} (\beta \oplus w_2^{(l)}(k_1,\dots,k_l),$$

$$\alpha \oplus \beta \oplus w_1^{(l)}(k_1,\dots,k_l) \oplus w_2^{(l)}(k_1,\dots,k_l)) \xrightarrow{f_{k_1}\dots f_{k_l}} (\alpha,\beta).$$

Утверждение доказано.

Пусть $g_k = f_{k_1}\dots f_{k_l}$, где преобразования f_{k_j} удовлетворяют условию утверждения 2 для всех $j \in \{\overline{1,l}\}$, $l \geq 1$. Пусть также (α_1,β_1), …, (α_t, β_t) – неизвестный открытый текст длины $t \geq 1$, $(\alpha_1^{(l)},\beta_1^{(l)}),\dots,(\alpha_t^{(l)},\beta_t^{(l)})$ – известный шифртескт, где $(\alpha_i^{(l)},\beta_i^{(l)}) = (\alpha_i,\beta_i)^{g_k}$, $i = \overline{1,t}$.

Из утверждения 2 следует, что без знания ключа по известному шифртексту $(\alpha_1^{(l)},\beta_1^{(l)}),\dots,(\alpha_t^{(l)},\beta_t^{(l)})$ можно получить следующую информацию об открытом тексте:

1) если $l \equiv 0 \pmod 3$, то $(\alpha,\beta) \xrightarrow{g_k^2} (\alpha,\beta)$ и $(\alpha_i^{(l)},\beta_i^{(l)}) \xrightarrow{g_k} (\alpha_i,\beta_i)$ для любого $i = \{\overline{1,t}\}$;

2) если $l \not\equiv 0 \pmod 3$, то $(\alpha,\beta) \xrightarrow{g_k^3} (\alpha,\beta)$ и $(\alpha_i^{(l)},\beta_i^{(l)}) \xrightarrow{g_k^2} (\alpha_i,\beta_i)$ для любого $i = \{\overline{1,t}\}$;

3) если $l \equiv 2 \pmod 3$, то $\beta_i^{(l)} \oplus \beta_j^{(l)} \sim_2 \alpha_i \oplus \alpha_j$ и $\beta_i^{(l)} \oplus \beta_j^{(l)} \oplus \alpha_i^{(l)} \oplus \alpha_j^{(l)} \sim_2 \beta_i \oplus \beta_j$ для любой пары $i, j \in \{\overline{1,t}\}$;

4) если $l \equiv 1 \pmod 3$, то $\alpha_i^{(l)} \oplus \alpha_j^{(l)} \sim_2 \beta_i \oplus \beta_j$, и $\beta_i^{(l)} \oplus \beta_j^{(l)} \oplus \alpha_i^{(l)} \oplus \alpha_j^{(l)} \sim_2 \alpha_i \oplus \alpha_j$ для любой пары $i, j \in \{\overline{1,t}\}$;

5) если $l \equiv 0 \pmod 3$, то $\alpha_i^{(l)} \sim_2 \alpha_i$, $\beta_i^{(l)} \sim_2 \beta_i$ для любого $i \in \{\overline{1,t}\}$.

Утверждение 3. Для любых подстановок $s = (s_1, s_2) \in S(V_m) \times S(V_m)$, $\pi \in S(V_m)$, справедливо равенство

$$(\alpha,\beta)^{f_\pi^s} = \left(\beta^{s_2^{-1}s_1}, \left(\beta^{s_2^{-1}\pi} \oplus \alpha^{s_1^{-1}}\right)^{s_2}\right).$$

Доказательство следует из равенств

$$(\alpha,\beta)^{f_\pi^s} = \left(\alpha^{s_1^{-1}},\beta^{s_2^{-1}}\right)^{f_{\pi^s}} = \left(\beta^{s_2^{-1}},\beta^{s_2^{-1}\pi} + \alpha^{s_1^{-1}}\right)^s = \left(\beta^{s_2^{-1}s_1}, \left(\beta^{s_2^{-1}\pi} + \alpha^{s_1^{-1}}\right)^{s_2}\right).$$

Следствие 2. В условиях утверждения 2, если $s_2^{-1}s_1 \in S_{2^{m-1}} \int S_2$ и $(\alpha+\beta)^{s_2} \sim_2 \alpha^{s_2} \oplus \beta^{s_2}$ для любых $\alpha, \beta \in V_m$, то

$$(\alpha,\beta) \xrightarrow{f_\pi^s} \left(\beta, \beta^{s_2^{-1}\pi s_2} \oplus \alpha\right).$$

В частности, если $s_2 \in S_{2^{m-1}} \int S_2$ или $s_2 \in GL_m$, то соотношение $(\alpha \oplus \beta)^{s_2} \sim_2 \alpha^{s_2} \oplus \beta^{s_2}$ справедливо для любых $\alpha, \beta \in V_m$.

Доказательство следует из утверждения 3.

ЛИТЕРАТУРА

1. *Schneier B.* Applied Cryptography, Protocols, Algorithms, and Source Code in C. Second edition. New York: John Wiley and Sons, 1996.
2. *Pieprzyk J.*, *Zhang X.M.* Permutation generators of alternating groups // AUSCRYPT'90. 1990, LNCS 453.
3. *Caranti A., Volta F.D., Sala M., Villani F.* Imprimitive permutations groups generated by the round functions of key-alternating block ciphers and truncated differential cryptanalysis // Workshop on Coding and Cryptography, UC Cork. 2005.
4. *Paterson K.G.* Imprimitive Permutation Groups and Trapdoors in Iterated Block Ciphers // FSE'99. 1999. LNCS 1636.
5. *Погорелов Б.А.*, *Пудовкина М.А.* Подметрики метрики Хемминга и преобразования, распространяющие искажения в заданное число раз // Труды по дискретной математике, АК РФ. 2007. Т. 10.
6. *Погорелов Б.А.*, *Пудовкина М.А.* Линейные структуры групп подстановок векторных пространств // Труды 3-й Междунар. конф. «Проблемы безопасности и противодействия терроризму, 2007». М.: МЦНМО, 2008.
7. *Фомичев В.М.* Дискретная математика и криптология. М.: ЗАО «Диалог-МИФИ», 2003.

2008 Математические методы криптографии № 2(2)

DOI 10.17223/20710410/2/13

УДК 518.6+681.3

АНАЛИЗ АТАК НА КВАНТОВЫЙ ПРОТОКОЛ ПЕРЕДАЧИ КЛЮЧА

В.Г. Скобелев

Институт прикладной математики и механики НАН Украины, г. Донецк

E-mail: skbv@iamm.ac.donetsk.ua

Исследуется вычислительная стойкость квантового протокола передачи ключа в предположении, что криптоаналитик управляет вероятностями выбора базисных векторов для измерения кубита, а также одновременным изменением базисов отправителя и адресата.

***Ключевые слова*:** *квантовая криптография*, *криптоанализ*, *активная атака*, *квантовый протокол передачи ключа.*

Применение квантовых вычислений к решению задач криптологии [1, 2] обосновывает актуальность исследования вычислительной стойкости квантовых алгоритмов. Возникает вопрос: что и как подвергается атаке? Сложность состоит в том, что в настоящее время недостаточно проработана формальная модель квантового компьютера, а искажение передаваемой информации за счет ее измерения приводит к новым типам атак, представляющим собой симбиоз пассивных и активных атак [3]. Поэтому естественно исследовать вычислительную стойкость квантовых алгоритмов решения конкретных, модельных задач криптологии. В настоящей работе в качестве такой задачи выбран анализ атак на классический квантовый протокол передачи ключа [4, 5].

Структура работы следующая: в п. 1 введены основные понятия и дана постановка задачи; в п. 2 исследуется атака на квантовый протокол передачи ключа в предположении, что криптоаналитик управляет только вероятностями выбора базисных векторов для измерения кубита. В п. 3 исследуется усиление этой атаки при условии, что криптоаналитик также может управлять одновременным изменением базисов отправителя и адресата. Заключение содержит ряд выводов.

1. Основные понятия и постановка задачи

Пусть $B = \{\boldsymbol{e}_0, \boldsymbol{e}_1\}$ − ортонормированный базис в 2-мерном комплексном векторном пространстве H. Кубит − это физическая система, состояние которой представлено вектором $|\xi\rangle = \lambda_0\boldsymbol{e}_0 + \lambda_1\boldsymbol{e}_1$, где $\lambda_0, \lambda_1 \in \mathbf{C}$ ($|\lambda_0|^2 + |\lambda_1|^2 = 1$). При измерении в базисе B кубит переходит в базисное состояние $\boldsymbol{e}_i$ ($i = 0, 1$) с вероятностью $|\lambda_0|^2$. В [4, 5] предложен следующий квантовый протокол передачи ключа (рис. 1, *а*).

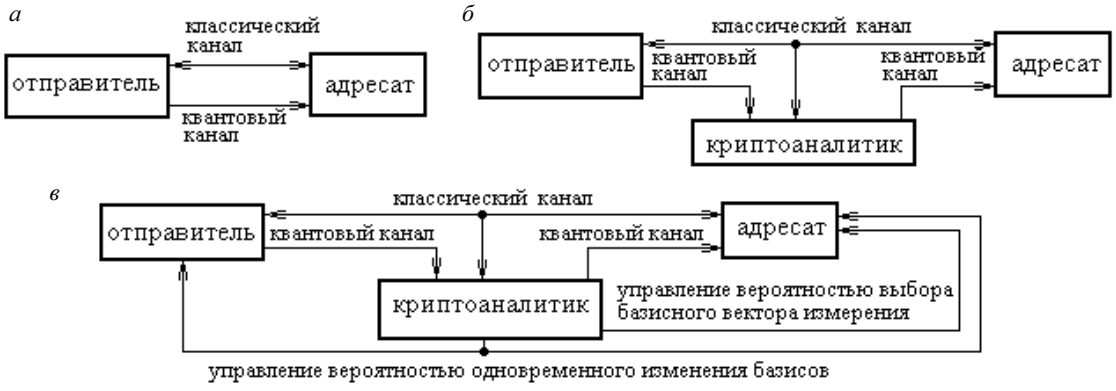

Рис. 1. Квантовый протокол передачи ключа: *а* – передача ключа при отсутствии атаки; *б* – передача ключа при классической атаке; *в* – передача ключа при исследуемых атаках

Отправитель и адресат располагают квантовым и классическим каналами: 1-й применяется для передачи ключа последовательностью кубитов, а 2-й − для контроля вычислений. Пусть $B_j = \{\boldsymbol{e}_0^{(j)}, \boldsymbol{e}_1^{(j)}\}$ ($j = 0, 1$) − такие ортонормированные базисы, что $\boldsymbol{e}_i^{(1)} = 2^{-0,5}(\boldsymbol{e}_0^{(0)} + (-1)^{1+i}\boldsymbol{e}_1^{(0)})$ ($i = 0, 1$). Значение i ($i = 0, 1$) бита кодируется в базисе B_j ($j = 0, 1$) вектором $\boldsymbol{e}_i^{(j)}$. Для каждого бита и отправитель и адресат случайным образом выбирают базис B_0 или B_1. По завершении процесса передачи последовательности отправитель и адресат со-

общают друг другу по классическому каналу, какие базисы были выбраны для кодирования и измерения каждого бита. Те биты, для которых базисы согласованы − ключ. Остальные биты отбрасываются. Доказано, что в среднем длина ключа составляет 50 % длины переданной последовательности.

Классическая атака на этот протокол состоит в следующем. Криптоаналитик перехватывает, измеряет передаваемые кубиты и пересылает измеренные кубиты адресату. Кроме того, криптоаналитик прослушивает классический канал (рис. 1, *б*). Доказано, что адресат в среднем верно измерит только 50 % от длины ключа. Поэтому, сравнив по открытому каналу некоторое число бит ключа, отправитель и адресат с соответствующей вероятностью обнаружат атаку. Отметим, что, хотя об этом нигде явно не сказано, такая атака основана на предположении о том, что адресат и криптоаналитик используют для измерения каждого кубита в каждом базисе B_j ($j = 0, 1$) один и тот же фиксированный базисный вектор $\boldsymbol{e}_i^{(j)}$.

Рассмотренная выше атака предполагает, что в распоряжении криптоаналитика имеется минимум средств. Усилим эту атаку за счет следующих предположений (рис. 1, *в*):

Предположение 1. Для измерения перехваченного кубита в базисе B_j ($j = 0, 1$) криптоаналитик выбирает базисный вектор $\boldsymbol{e}_i^{(j)}$ ($i = 0, 1$) с вероятностью $p_1^{(j)}(i)$.

Предположение 2. Криптоаналитик определяет вероятность $p_2^{(j)}(i)$ ($i, j \in \{0, 1\}$) выбора адресатом базисного вектора $\boldsymbol{e}_i^{(j)}$ при измерении кубита в базисе B_j, причем адресат не располагает информацией о том, что у него произошло изменение базисного вектора.

Предположение 3. Криптоаналитик может одновременно изменять у отправителя и адресата базис B_j ($j = 0, 1$) на базис B_{1-j} с вероятностью $p_0(j)$, причем ни отправитель, ни адресат не располагают информацией о том, что у них произошло изменение базиса.

Из предположений 1 и 2 вытекает, что $p_k^{(j)}(1 - i) = 1 - p_k^{(j)}(i)$ для всех $i, j \in \{0, 1\}$ и $k \in \{1, 2\}$.

2. Атака при управлении вероятностями выбора базисных векторов для измерения кубита

Рассмотрим атаку на квантовый протокол передачи ключа, определяемую предположениями 1 и 2.

Пусть $P_{jhj}^{(i)}$ ($i, h, j \in \{0, 1\}$) − вероятность правильного считывания адресатом значения i-го бита при условии, что для данного кубита отправитель и адресат используют базис B_j, а криптоаналитик − базис B_h. Из анализа процесса передачи бита при согласованных базисах отправителя и адресата, представленного на рис. 2 и 3, вытекает

Теорема 1. Истинны равенства

$$P_{jjj}^{(i)} = 1 - p_2^{(j)}(i) + p_1^{(j)}(i) \cdot p_2^{(j)}(i) \quad (j = 0, 1),$$

$$P_{j,1-j,j}^{(i)} = 0{,}75 - 0{,}5\, p_2^{(j)}(i) \quad (j = 0, 1).$$

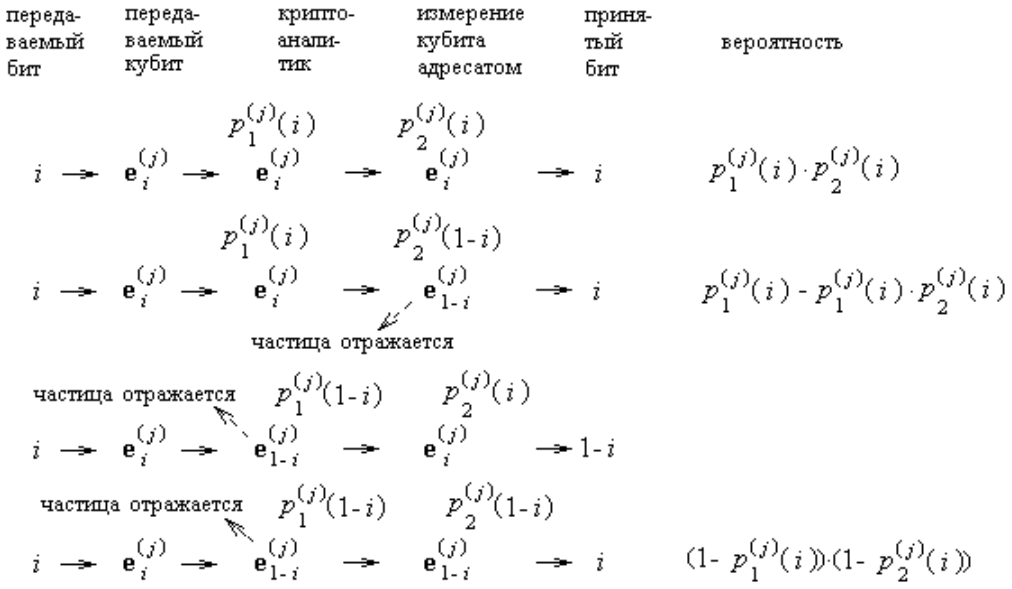

Рис. 2. Процесс передачи значения i ($i = 0, 1$) бита при условии, что для данного кубита отправитель, криптоаналитик и адресат используют базис B_j ($j = 0, 1$)

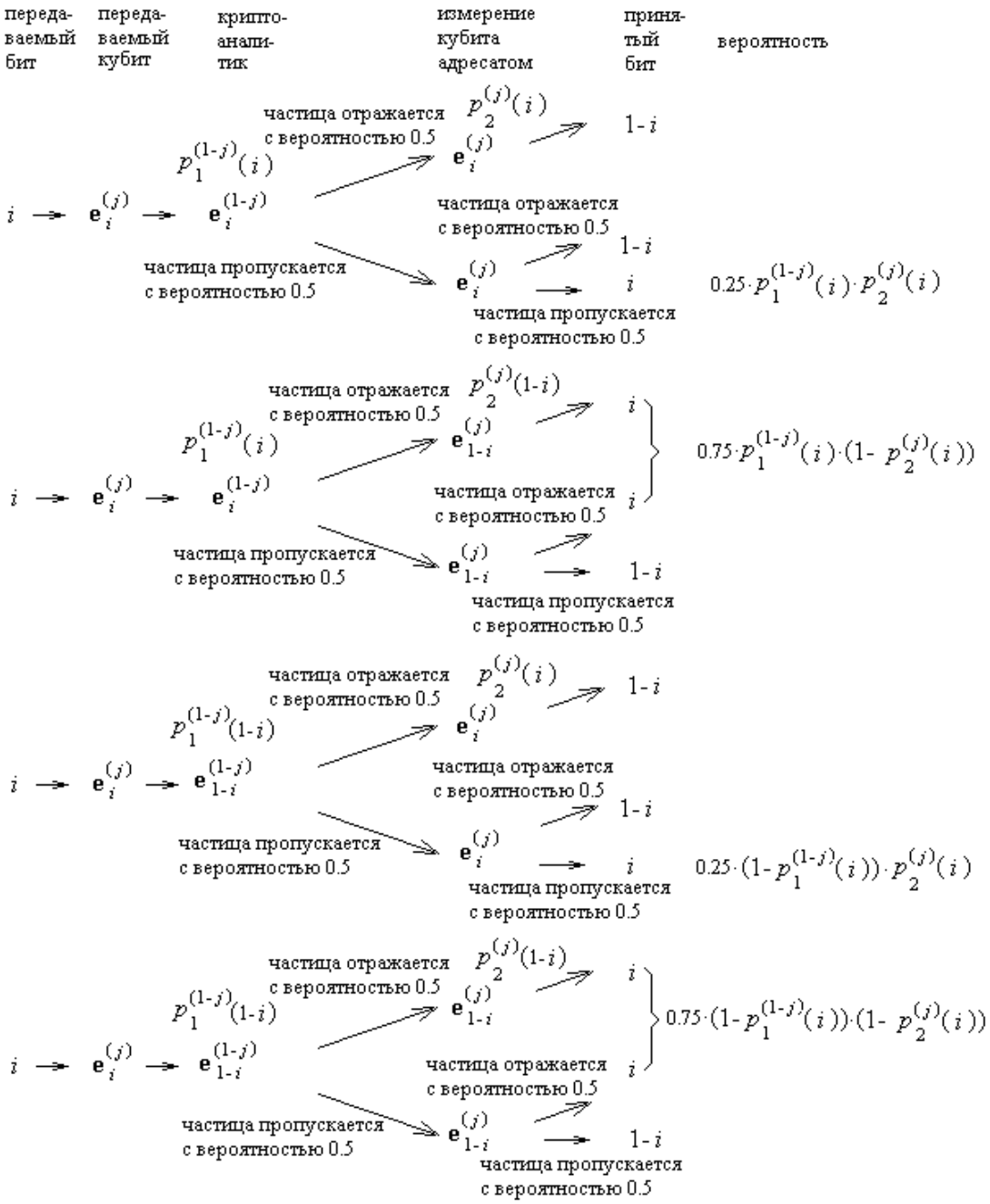

Рис. 3. Процесс передачи значения i (i = 0, 1) бита при условии, что для данного кубита отправитель и адресат используют базис B_j (j = 0, 1), а криптоаналитик − базис B_{1-j}

Обозначим через $P_{jhj}(\alpha)$ ($j, h \in \{0, 1\}$, $\alpha \in [0, 1]$) вероятность правильного считывания адресатом значения бита при условии, что для данного кубита отправитель и адресат используют базис B_j , криптоаналитик − базис B_h, а вероятность пересылки символа 0 отправителем равна α. Из теоремы 1 вытекает

Теорема 2. Для всех $\alpha \in [0 , 1]$ истинны равенства

$$P_{jjj}(\alpha) = 1 - p_1^{(j)}(0) + p_1^{(j)}(0) \cdot p_2^{(j)}(0) + \alpha\,(p_1^{(j)}(0) - p_2^{(j)}(0))\ \ (j = 0, 1); \tag{1}$$

$$P_{j,1-j,j}(\alpha) = 0{,}25 + 0{,}5\alpha + (0{,}5 - \alpha) \cdot p_2^{(j)}(0)\ \ (j = 0, 1). \tag{2}$$

Отметим ряд следствий из равенства (1).

I. Пусть $p_2^{(j)}(0) = 0{,}5$ (j = 0, 1). Тогда

$$P_{jjj}(\alpha) = 1 - (0{,}5 - \alpha) \cdot p_1^{(j)}(0) - 0{,}5\alpha\ \ (j = 0, 1). \tag{3}$$

При этом из (3) вытекает, что для всех $\alpha \in [0, 1]$

$$P_{jjj}(\alpha) = \begin{cases} 1-0,5\cdot\alpha, & \text{если } p_1^{(j)}(0)=0 \\ 0,5\cdot(1+\alpha), & \text{если } p_1^{(j)}(0)=1 \end{cases} \quad (j=0,\,1).$$

II. Пусть $p_1^{(j)}(0) = p_2^{(j)}(0) = p^{(j)}(0)$ $(j = 0, 1)$. Тогда

$$P_{jjj}(\alpha) = 1 - p^{(j)}(0) + (p^{(j)}(0))^2 \quad (j=0,\,1), \tag{4}$$

т. е. вероятность $P_{jjj}(\alpha)$ не зависит от вероятности α. Из (4) вытекает, что $P_{jjj}(\alpha) \in [0{,}75,\,1]$ $(j = 0, 1)$ для всех $p^{(j)}(0) \in [0;1]$, причем

$$P_{jjj}(\alpha) = \begin{cases} 0,75, & \text{если } p^{(j)}(0)=0,5 \\ 1, & \text{если } p^{(j)}(0)\in\{0;1\} \end{cases} \quad (j=0\,,\,1)\,.$$

III. Пусть $p_1^{(j)}(0) \neq p_2^{(j)}(0)$ $(j = 0, 1)$. Тогда

1. Для области значений вероятности $P_{jjj}(\alpha)$ $(j = 0, 1)$, как функции от вероятности α, истинно равенство

$$\mathrm{Val}\, P_{jjj}(\alpha) = [l_1, L_1] \quad (j=0,\,1),$$

где

$$l_1 = \min\{1 - p_1^{(j)}(0) + p_1^{(j)}(0)\cdot p_2^{(j)}(0),\, 1 - p_2^{(j)}(0) + p_1^{(j)}(0)\cdot p_2^{(j)}(0)\} \quad (j=0,\,1); \tag{5}$$

$$L_1 = \max\{1 - p_1^{(j)}(0) + p_1^{(j)}(0)\cdot p_2^{(j)}(0),\, 1 - p_2^{(j)}(0) + p_1^{(j)}(0)\cdot p_2^{(j)}(0)\} \quad (j=0,\,1). \tag{6}$$

2. Из (5) и (6) вытекает, что для всех значений $\alpha \in [0, 1]$, если либо $p_1^{(j)}(0) \to 0$, $p_2^{(j)}(0) \to 0$ $(j = 0, 1)$, либо $p_1^{(j)}(0) \to 1$, $p_2^{(j)}(0) \to 1$ $(j = 0, 1)$, то $l_1 \to 1$ и $L_1 \to 1$, т. е. $P_{jjj}(\alpha) \to 1$ $(j = 0, 1)$ для всех $\alpha \in [0, 1]$.

3. Вероятность $P_{jjj}(\alpha) \to 1$ $(j = 0, 1)$ − монотонная функция от вероятности α, причем $P_{jjj}(\alpha)$ монотонно возрастает, если $p_1^{(j)}(0) > p_2^{(j)}(0)$, и монотонно убывает, если $p_1^{(j)}(0) < p_2^{(j)}(0)$.

4. Пусть $p_1^{(j)}(0) \to 1$, $p_2^{(j)}(0) \to 0$ $(j = 0, 1)$. Тогда $P_{jjj}(\alpha) \to \alpha$ $(j = 0, 1)$ для всех $\alpha \in [0, 1]$.

5. Пусть $p_1^{(j)}(0) \to 0$, $p_2^{(j)}(0) \to 1$ $(j = 0, 1)$. Тогда $P_{jjj}(\alpha) \to 1 - \alpha$ $(j = 0, 1)$ для всех $\alpha \in [0, 1]$.

Отметим ряд следствий из равенства (2).

I. Вероятность $P_{j,1-j,j}(\alpha)$ $(j = 0, 1)$ вообще не зависит от вероятности выбора криптоаналитиком базисного вектора для измерения перехваченного кубита.

II. Для всех $\alpha \in [0, 1]$

$$P_{j,1-j,j}(\alpha) = \begin{cases} 0,25+0,5\cdot\alpha, & \text{если } p_2^{(j)}(0)=0 \\ 0,75-0,5\cdot\alpha, & \text{если } p_2^{(j)}(0)=1 \end{cases} \quad (j=0\,,\,1)\,.$$

III. Пусть $p_2^{(j)}(0) = 0{,}5$. Тогда $P_{j,1-j,j}(\alpha) = 0{,}5$ $(j = 0, 1)$, т. е. вероятность $P_{j,1-j,j}(\alpha)$ не зависит от α.

IV. Пусть $p_2^{(j)}(0) \neq 0{,}5$. Тогда

1. Для области значений вероятности $P_{j,1-j,j}(\alpha)$ $(j = 0, 1)$, как функции от α, истинно равенство

$$\mathrm{Val}\, P_{jjj}(\alpha) = [l_2\,,\, L_2] \quad (j=0,\,1),$$

где

$$l_2 = \min\{0{,}25 + 0{,}5\cdot p_2^{(j)}(0),\, 0{,}75 - 0{,}5\, p_2^{(j)}(0)\} \quad (j=0,\,1); \tag{7}$$

$$L_2 = \max\{0{,}25 + 0{,}5\cdot p_2^{(j)}(0),\, 0{,}75 - 0{,}5\, p_2^{(j)}(0)\} \quad (j=0,\,1). \tag{8}$$

2. Вероятность $P_{j,1-j,j}(\alpha)$ $(j = 0, 1)$ − монотонная функция от вероятности α, причем $P_{j,1-j,j}(\alpha)$ монотонно возрастает, если $p_2^{(j)}(0) < 0{,}5$, и монотонно убывает, если $p_2^{(j)}(0) > 0{,}5$.

3. Из (7) и (8) вытекает, что для всех значений $\alpha \in [0, 1]$, если $p_2^{(j)}(0) \to 0{,}5$, то $l_2 \to 0{,}5$ и $L_2 \to 0{,}5$, т. е.

$$P_{j,1-j,j}(\alpha) \to 0{,}5 \quad (j=0,\,1).$$

Пусть $P_1(\alpha)$ $(\alpha \in [0, 1])$ − вероятность правильного считывания адресатом значения бита при условии, что для данного кубита базисы отправителя и адресата согласованы, а вероятность пересылки символа 0 отправителем равна α. Тогда

$$P_1(\alpha) = 0{,}125(P_{000}(\alpha) + P_{111}(\alpha) + P_{010}(\alpha) + P_{101}(\alpha)). \tag{9}$$

Из теоремы 2 и равенства (9) вытекает

Теорема 3. Для всех $\alpha \in [0, 1]$ истинно равенство

$$P_1(\alpha) = 0{,}125(\,2{,}5 + \alpha - (1-\alpha)\cdot(p_1^{(0)}(0) + p_1^{(1)}(0)) +$$

$$+ (0{,}5 - 2\alpha)\cdot(p_2^{(0)}(0) + p_2^{(1)}(0)) + p_1^{(0)}(0)\cdot p_2^{(0)}(0) + p_1^{(0)}(0)\cdot p_2^{(0)}(0))\,.$$

Отметим ряд следствий из равенства (9).

I. Если $p_1^{(j)}(0) \to 0$, $p_2^{(j)}(0) \to 0$ ($j = 0, 1$), то $P_1(\alpha) \to 0{,}3125 + 0{,}125\alpha$. При этом:

1) если $\alpha \to 0$, то $P_1(\alpha) \to 0{,}3125$;
2) если $\alpha \to 0{,}5$, то $P_1(\alpha) \to 0{,}3750$;
3) если $\alpha \to 1$, то $P_1(\alpha) \to 0{,}4375$.

II. Если $p_1^{(j)}(0) \to 1$, $p_2^{(j)}(0) \to 0$ ($j = 0, 1$), то $P_1(\alpha) \to 0{,}0625 + 0{,}3750\alpha$. При этом:

1) если $\alpha \to 0$, то $P_1(\alpha) \to 0{,}0625$;
2) если $\alpha \to 0{,}5$, то $P_1(\alpha) \to 0{,}2500$;
3) если $\alpha \to 1$, то $P_1(\alpha) \to 0{,}4375$.

Если $p_1^{(j)}(0) \to 1$, $p_2^{(j)}(0) \to 1$ ($j = 0, 1$), то $P_1(\alpha) \to 0{,}4375 - 0{,}125\alpha$. При этом:

1) если $\alpha \to 0$, то $P_1(\alpha) \to 0{,}4375$;
2) если $\alpha \to 0{,}5$, то $P_1(\alpha) \to 0{,}3750$;
3) если $\alpha \to 1$, то $P_1(\alpha) \to 0{,}3125$.

IV. Если $p_1^{(j)}(0) \to 0{,}5$, $p_2^{(j)}(0) \to 0{,}5$ ($j = 0, 1$), то $P_1(\alpha) = 0{,}3125$.

Проведенный анализ показывает, что значение вероятности $P_1(\alpha)$ ($\alpha \in [0, 1]$) колеблется в достаточно широких пределах, по крайней мере, в промежутке [0,2500, 0,4375], в зависимости от значений вероятностей $p_1^{(j)}(0)$, $p_2^{(j)}(0)$ ($j = 0, 1$) и α. Это означает, что криптоаналитик располагает достаточно широкими возможностями для того, чтобы существенно затруднить процесс обнаружения его действий. Действительно, если генератор последовательностей отправителя далек от псевдослучайного, то в среднем до 87,5% ключа может быть передано верно. Даже при наличии случайного генератора последовательностей у отправителя в среднем до 75% ключа может быть передано верно.

3. Атака при управлении изменением базисов отправителя и адресата

Рассмотрим теперь атаку на квантовый протокол передачи ключа, определяемую предположениями 1 – 3.

Пусть $P_2(\alpha)$ ($\alpha \in [0, 1]$) − вероятность правильного считывания адресатом значения бита при условии, что для данного кубита базисы отправителя и адресата согласованы, вероятность пересылки символа 0 отправителем равна α, а вероятность одновременного изменения криптоаналитиком у отправителя и адресата базиса B_j ($j = 0,1$) на базис B_{1-j} равна $p_0(j)$. Тогда имеет место

Теорема 4. Для всех $\alpha \in [0, 1]$ истинно равенство

$$P_2(\alpha) = 0{,}125(P_{000}(\alpha) + P_{111}(\alpha) + P_{010}(\alpha) + P_{101}(\alpha)) + \\ + (p_0(1) - p_0(0))(P_{000}(\alpha) + P_{010}(\alpha) - P_{111}(\alpha) - P_{101}(\alpha))). \qquad (10)$$

Из равенства (10) вытекает, что:

1) неравенство $P_2(\alpha) > P_1(\alpha)$ истинно тогда и только тогда, когда либо $P_{000}(\alpha) + P_{010}(\alpha) - P_{111}(\alpha) - P_{101}(\alpha) < 0$ и $p_0(1) - p_0(0) < 0$, либо $P_{000}(\alpha) + P_{010}(\alpha) - P_{111}(\alpha) - P_{101}(\alpha) < 0$ и $p_0(1) - p_0(0) > 0$;

2) равенство $P_2(\alpha) > P_1(\alpha)$ истинно тогда и только тогда, когда $P_{000}(\alpha) + P_{010}(\alpha) - P_{111}(\alpha) - P_{101}(\alpha) = 0$ или $p_0(1) = p_0(0)$.

Таким образом, показано, что дополнительная возможность криптоаналитика управлять одновременным изменением базисов отправителя и адресата может усилить его атаку на квантовый протокол передачи ключа.

Заключение

Показано, что расширение возможностей криптоаналитика за счет управления вероятностями выбора базисных векторов для измерения кубита, а также одновременным изменением базисов отправителя и адресата может значительно усилить его атаку на квантовый протокол передачи ключа. Показано, что эффективность атаки существенно зависит от генератора последовательностей, имеющегося у отправителя. Более тонкий анализ этой зависимости − одно из возможных направлений дальнейших исследований. Второе направление исследований связано с анализом зависимости вероятности $P_2(\alpha)$ от значений вероятностей $p_1^{(j)}(i)$, $p_2^{(j)}(i)$, $p_0(j)$ ($i, j \in \{0, 1\}$) и α.

ЛИТЕРАТУРА

1. *Ожигов Ю.И.* Квантовые вычисления. М.: МГУ, 2003.
2. *Нильсен М., Чанг И.* Квантовые вычисления и квантовая информация. М.: Мир, 2006.
3. *Алферов А.П. и др.* Основы криптографии. М.: Гелиос АРВ, 2002.
4. *Bennett C.H., Brassard G.* Quantum public key distribution system // IBM Techn. Disclosure Bulletin. 1985. V. 28. P. 3153 – 3164.
5. *Bennett C.H., Brassard G.* Quantum public key distribution reinvented. 1987. SIGACTN: SIGACT News. 18.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

DOI 10.17223/20710410/2/14

УДК 681.3

МЕТОДЫ ЗАЩИТЫ РЕЧЕВОЙ ИНФОРМАЦИИ

А.М. Гришин

Институт криптографии, связи и информатики Академии ФСБ России, г. Москва

E-mail: av123470@comtv.ru

В статье рассматриваются основные задачи, которые возникают при построении системы защиты речевых сигналов, и даются рекомендации по их решению.

Ключевые слова: *защита речи, криптографические методы защиты.*

Человеческая речь, и в частности телефонные переговоры, остается важнейшим каналом информационного взаимодействия. Зачастую развитие и введение в эксплуатацию новых систем связи направлено на совершенствование именно этого метода общения. Одновременно усиливается потребность в обеспечении конфиденциальности речевого обмена и защите информации, имеющей речевую природу.

В настоящий момент разработан достаточно широкий арсенал различных средств защиты (формальных и неформальных), которые могут обеспечить требуемый уровень защищенности разного рода информации, в том числе и речевой. Развитие неформальных средств защиты (законодательных, организационных, морально-этических и т.п.) проводится в рамках общего законодательного процесса и посредством совершенствования соответствующих инструкций.

В России сложилась достаточно разветвленная правовая система, которая регламентирует многие аспекты организации и обеспечения информационной безопасности [1]. Важное место в этой системе занимают требования по лицензированию и сертификации, но возможность применения этих требований к защите собственных информационных ресурсов в собственных же интересах не очевидна [2]. Есть определенные правовые коллизии и в широком применении ряда криптографических средств, строго говоря, не прошедших сертификацию в России, но использующихся в глобальных системах связи.

Причины такого положения, видимо, кроются в необходимости применения различных критериев, в том числе и правовых, в вопросах сертификации коммерческих систем связи (требования по защите информации в коммерческих целях) и систем связи специального назначения (требования по защите гостайны).

На развитие и совершенствование арсенала технических средств защиты речевой информации оказывают влияние многочисленные объективные и субъективные факторы, основные из которых сформулированы ниже.

F1. Речевой и слуховой аппарат человека является прекрасно сопряженной и исключительно помехоустойчивой системой. Поэтому подавление смыслового восприятия речи происходит при отношении шум/сигнал в несколько сотен процентов, а подавление признаков речи (т.е. невозможность фиксации факта разговора) достигается при отношении шум/сигнал ≈ 10 и выше [2, 3].

F2. Аппаратура и системы связи, связанные с обработкой и передачей речевой информации, постоянно совершенствуются и развиваются. Для мобильных телефонов и наладочных компьютеров речевой интерфейс является самым удобным способом обмена информацией. Соответствующие изменения затрагивают как возможные каналы утечки речевой информации, так и методы получения несанкционированного доступа (НСД) к этой информации. Эти процессы требуют адекватной реакции при разработке стратегии защиты и совершенствовании методов защиты речевых сигналов.

F3. Широкое распространение получают принципиально новые автоматизированные и компьютеризованные системы обработки, в которых происходит обработка, накопление и хранение огромных массивов информации, в том числе имеющих речевую природу (записи переговоров, речевая почта, данные акустического контроля и т.п.). В связи с этим требуется разработка технологий и методов защиты речевой информации, передача которой по каналам связи не предполагается.

F4. Постоянно развиваются методы и совершенствуется аппаратура для получения несанкционированного доступа к речевой информации, в частности к телефонным переговорам. В силу своей специфики и протяженности системы связи, предоставляющие услуги телефонных переговоров и речевого общения, являются наиболее уязвимыми для НСД и утечки конфиденциальной информации.

F5. Интеграция России в мировую экономическую систему и динамичное развитие бизнеса, который по своей природе стремится формировать и заполнять имеющиеся пробелы в сфере услуг, приводят к появлению хорошо оснащенных фирм, имеющих значительные технические возможности по НСД к конфиденциальной информации. Это, в свою очередь, меняет модель противника – один из важнейших параметров, которые необходимо рассматривать при разработке мер защиты.

Традиционно рассматривают две основных задачи, которые необходимо решить для предотвращения утечки конфиденциальной речевой информации.

Z1. Задача обеспечения безопасности переговоров в помещении или в пределах контролируемой территории.

Z2. Задача обеспечения защиты речевой информации в канале связи.

Основные факторы, перечисленные выше, позволяют говорить по крайней мере еще о двух направлениях, по которым необходима организация специальных мероприятий и мер защиты.

Z3. Обеспечение постоянного контроля эффективности защиты речевой информации с целью предотвращения появления новых каналов утечки при кажущемся достаточным уровне защиты.

Z4. Накопление и хранение в защищенном виде массивов различной информации, имеющей речевую природу. Сюда же, видимо, следует отнести и информацию мультимедийного характера.

Для решения задачи Z4 можно применять стандартные методы, позволяющие накапливать и хранить в защищенном виде информацию конфиденциального характера. Но специфика объекта защиты и требования к работе с записями речевых переговоров вынуждают рекомендовать использовать для этих целей отдельные защищенные помещения, вычислительные средства и специальные информационно-справочные и информационно-поисковые системы.

Каналы телефонной связи являются наиболее уязвимыми с точки зрения организации НСД к конфиденциальной информации. Контролировать телефонные переговоры можно на всем протяжении телефонной линии, а при использовании мобильной связи еще и во всей зоне распространения радиосигнала [4 – 6].

В настоящее время можно говорить о следующих видах телефонной связи:

- стандартная телефонная связь, которая осуществляется по коммутируемым каналам;
- мобильная связь, основным примером которой можно считать связь по стандарту GSM;
- цифровая телефония (IP-телефония), которая осуществляется по сетям с коммутацией пакетов.

Каждый вид телефонной связи имеет свои особенности, которые необходимо учитывать при построении концепции защиты информации.

Штатная концепция защиты речевых переговоров при стандартной телефонной связи состоит в предположении отсутствия у злоумышленника доступа к телефонным каналам. Каких-либо средств защиты данная система телефонной связи не предусматривает. При отсутствии уверенности в такой «системе» защиты решение задачи обеспечения безопасности переговоров полностью ложится на абонентов.

В основе концепции защиты информации в системе связи стандарта GSM лежат криптографические протоколы аутентификации, алгоритмы шифрования трафика в радиоканале и система временных идентификаторов абонентов [7]. Все эти средства защиты обеспечиваются самой системой связи.

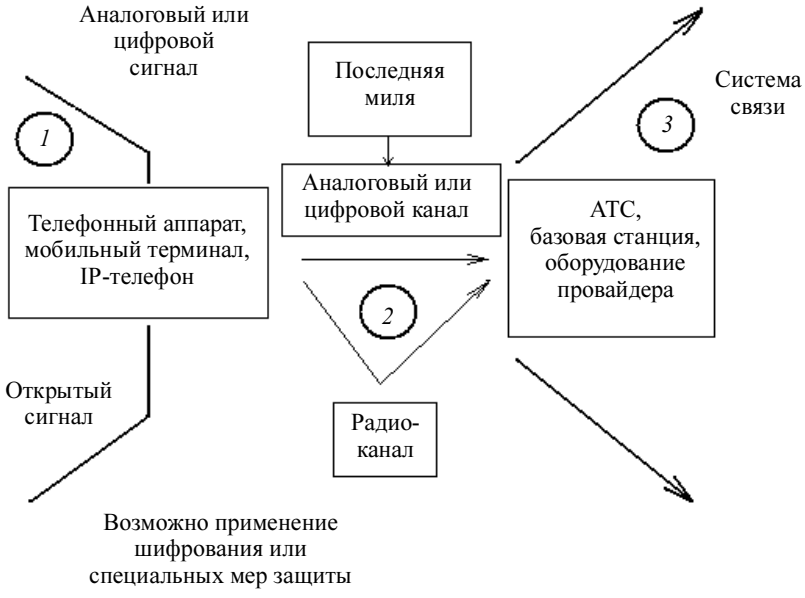

Рис.1. Общая модель телефонной связи

Цифровая телефония допускает применение практически всего спектра средств криптозащиты (защищенные протоколы, шифрование трафика и т.п.), причем это может обеспечиваться как штатными средствами защиты системы связи (провайдера), так и оборудованием абонентов.

Для пользователя все три вида телефонного обслуживания представляются как единая телефонная сеть, и он зачастую не знает, как точно осуществляется то или иное телефонное соединение. Поэтому логично для рассмотрения вопросов обеспечения безопасности схематично представить укрупненную модель телефонной связи (рис. 1).

Цифрами обозначены «точки» (места), в которых условия доступа к речевым сигналам с целью НСД имеют принципиальные отличия.

Т о ч к а *1*. Помещение, пространство на улице и т.п., в котором абонент непосредственно осуществляет телефонное общение.

Данная точка характеризуется следующими основными особенностями:

• наличием открытого речевого сигнала (не зашифрованного) в аналоговой форме;

• при телефонном разговоре имеется (слышен) сигнал только одного абонента;

• имеются определенные ограничения по возможностям применения средств защиты (средства как минимум не должны мешать ведению переговоров), невозможно использовать криптографические методы защиты.

Т о ч к а *2*. Канал связи – аналоговый, цифровой или радиоканал – между терминалом абонента и оборудованием системы связи. Для стандартной телефонной связи это АТС. Для мобильной связи – базовая станция. Для IP-телефонии – оборудование провайдера.

Точка характеризуется:

• в определенной степени постоянным и достаточно стабильным каналом связи, который невозможно обеспечить физической защитой на всем протяжении;

• сигнал может иметь аналоговую или цифровую форму, быть открытым или зашифрованным;

• в коммутируемом канале связи присутствуют одновременно сигналы обоих абонентов [8];

• можно использовать практически любые средства защиты, включая криптографические протоколы аутентификации и многоуровневое шифрование [2, 6, 9, 10].

Т о ч к а *3*. Оборудование и каналы той или иной системы связи.

Главная цель выделения точки *3* состоит в необходимости подчеркнуть факт, что условия осуществления НСД к телефонным переговорам «внутри» системы связи имеют место быть, и они могут принципиально отличаться от условий осуществления НСД на «последней» миле (в точке *2*). Причем эти условия могут быть как значительно проще, так и значительно сложнее. Но в любом случае для осуществления НСД в точке *3* необходимо иметь доступ к штатному оборудованию системы связи (оборудованию провайдера).

В точке *1* необходимо обеспечить решение задач Z1 и Z3.

Задача защиты переговоров, происходящих в помещении или на контролируемой территории, всегда может быть решена ценой определенных затрат и с созданием больших или меньших неудобств для переговаривающихся персон [2, 10]. Это обеспечивается:

- проверкой помещения и определенным контролем прилегающей территории, использованием технических средств (розеток, телефонных аппаратов, оргтехники и т.п.), исключающих утечку информации по побочным каналам;

- организацией соответствующего режима доступа в проверенное и контролируемое помещение;

- применением средств физической защиты информации, включающих в себя постановщики заградительных помех, нейтрализаторы, фильтры и средства физического поиска каналов утечки информации. Причем желательно обеспечить создание некоррелированных помех, исключающих возможность их компенсации при многоканальном съеме информации [11];

- постоянным мониторингом и оценкой качества защиты речевой информации на объекте. Существует много объективных и субъективных причин, которые могут явиться источником сбоев и нарушений в функционировании систем защиты в рабочих помещениях.

Очевидно, приведенная выше система мер в основном направлена на обеспечение безопасности связи со стационарных телефонов (в том числе и IP) и на предотвращение утечки по побочным каналам, одной из причин которых может являться телефон мобильной связи. Безопасность телефонных переговоров вне контролируемого помещения и в мобильном варианте эта система мер не обеспечивает.

Для предотвращения НСД к речевой информации в точке *2* можно использовать практически любые технические средства. В частности, для защиты обычных телефонных каналов сегодняшний рынок представляет пять разновидностей специальной техники [10, 12]:

- анализаторы телефонных линий;

- средства пассивной защиты;

- постановщики активной заградительной помехи;

- односторонние маскираторы речи [13];

- криптографические системы защиты [6, 9, 14].

Предназначение технических средств, относящихся к первым трем группам, достаточно очевидно.

Принято выделять три типа устройств [15], обеспечивающих криптографическую защиту речевой информации: маскираторы, скремблеры и устройства с передачей шифрованной речи в цифровом виде. Маскираторы и скремблеры относятся к аппаратуре временной стойкости [2, 9], так как используют передачу преобразованного сигнала по каналу связи в аналоговом виде. Вообще, провести строгое обоснование степени защищенности скремблеров крайне сложно.

Для гарантированной защиты телефонных переговоров желательно использовать аппаратуру, построенную на принципах цифровой передачи речи и обеспечивающую криптографическую защиту на всех этапах передачи.

Таким образом, оба абонента телефонной связи должны быть оснащены соответствующей шифртехникой, что является определенным неудобством. Вторым важным недостатком является тот факт, что в настоящее время ни один из скремблеров [12] не имеет надежной системы предотвращения перехвата речевой информации из помещения по телефонной линии, находящейся в отбое. Следовательно, такая аппаратура предоставляет принципиальную возможность проводить НСД в точке *1* (см. рис. 1) по техническим каналам утечки: акустическому, электромагнитному, сетевому и др.

В какой-то степени решить вопросы защиты речевого обмена в точке *2* позволяют односторонние маскираторы [13], но говорить о полной, надежной и доказательной защите информации в этом случае нет оснований.

Для защиты в точке *2* сигналов IP-телефонии из приведенного выше списка специальной техники можно использовать анализаторы телефонных линий (для контроля над возможными несанкционированными подключениями к линии) и цифровые системы криптографической защиты. Применение технических средств, вносящих помехи в канал связи, приведет к разрушению цифрового канала и невозможности использования IP-телефонии.

Как видно из рис. 1, концепция защиты информации сотовых систем, по сути, ограничивается только точкой *2* (т.е. радиоканалом). О мерах по дальнейшей защите должны позаботиться сами абоненты. Решить эти вопросы можно путем применения специальных криптографических средства абонентского шифрования, которые позволяют защищать речевой сигнал на всем пути следования от одного мобильного терминала до другого.

Применение подобных криптографических средств позволяет защищать речевую информацию в телефонных проводах, системах связи IP-телефонии и сотовых сетях. По сути, это единственная возможность построения надежной (и доказательной) системы защиты речевых переговоров в точках *2* и *3*.

Таким образом, надежное перекрытие возможных каналов утечки в охраняемых помещениях и применение сертифицированных криптографических средств, позволяющих шифровать информацию на всем протяжении линий связи между абонентами, позволяет построить надежную систему защиту для конфиденциального обмена речевой информацией. Правомерность таких рекомендаций подтверждают и некоторые публикации [16], в которых рассматриваются зарубежные технологии и терминология доступа к конфиденциальной информации. Доступ к данным в точке *1* характеризуется как доступ к открытой информации – «информации в покое» (information at rest). В противоположном состоянии – «информация в движении» (info in motion), открытый текст может быть зашифрован сильным криптоалгоритмом, и быстро подступиться к нему уже невозможно.

ЛИТЕРАТУРА

1. *Развитие* правового обеспечения информационной безопасности / Под ред. А.А. Стрельцова. М.: Престиж, 2006.
2. *Кравченко В.Б.* Защита речевой информации в каналах связи // Специальная техника. 1999. № 4. С. 2 – 9; 1999. № 5. С. 2 – 11.
3. *Цвикер Э.*, *Фельдкеллер Р.* Ухо как приемник информации / Пер. под общ. ред. Б.Г. Белкина. М.: Связь, 1971.
4. *Закрытие* телефонных переговоров. ВЕБ форум по безопасности. http://www.sec.ru/
5. *Материалы* сайта http://www.Phreaking.RU/
6. *Sutton R.J.* Secure Communications: Applications and Management. John Wiley & Sons, 2002.
7. *Ратынский М.* Телефон в кармане. Путеводитель по сотовой связи. М.: Радио и связь, 2000.
8. *Лагутенко О.И.* Модемы: Справочник пользователя. СПб.: Лань, 1997.
9. *Алферов А.П.*, *Зубов А.Ю.*, *Кузьмин А.С.*, *Черемушкин А.В.* Основы криптографии. М.: Гелиос АРВ, 2001.
10. *Петраков А.В*. Основы практической защиты информации. М.: Радио и связь, 1999.
11. *Бортников А.Н.*, *Губин С.В.*, *Комаров И.В.*, *Майоров В.И.* Совершенствование технологий информационной безопасности речи // Конфидент. 2001. № 4.
12. *Сталенков С.* Способы и защита телефонных линий. http://daily.sec.ru/
13. *Абалмазов Э.И.* Новая технология защиты телефонных переговоров // Специальная техника. 1998. № 1. С. 3 – 9.
14. *Beker H.J.*, *Piper F.C.* Secure Speech Communications. London: Academic Press, 1986.
15. *Смирнов В.* Защита телефонных переговоров // Банковские технологии. 1996. № 8. С. 5 – 11.
16. *Берд К.* Искусство быть // Компьютерра. 2005. №11. http://www.computerra.ru/offline/2005/583/38052/

2008 Математические основы компьютерной безопасности № 2(2)

DOI 10.17223/20710410/2/15

УДК 004.056

ФУНДАМЕНТАЛЬНЫЕ ОСНОВЫ МЕТОДИК БАЗОВОГО ЭКСПЕРТНОГО АНАЛИЗА ИНФОРМАЦИОННЫХ РИСКОВ[1]

В.В. Золотарев, Е.А. Ширкова

Сибирский государственный аэрокосмический университет, *г. Красноярск*

E-mail: amida@land.ru

В работе сделана попытка анализа процедуры формирования экспертной оценки информационного риска, целью которого явилось выявление фундаментальных основ базовых методик экспертного анализа информационных рисков. Представленные результаты получены на основе теоретико-множественного и графоаналитического подходов.

Ключевые слова: *управление риском*, *информационный риск*, *экспертные оценки.*

Практические подходы, формируемые при решении задач анализа рисков, часто не имеют хорошо проработанного формального обоснования, математического аппарата и других характеристик, необходимых для анализа проблемы. Кроме того, при разработке новых методик базового экспертного анализа информационных рисков и модификации существующих опираются не на фундаментальные результаты в предметной области, а на особенности практических задач.

В работе представлены результаты исследования, направленного на решение задачи обобщения и анализа основ методик экспертного анализа, применяемых в предметной области.

Полученные результаты свидетельствуют о необходимости пересмотра некоторых положений, широко используемых при формировании методик базового экспертного анализа информационных рисков.

Для анализа использованы исходные данные по применяемым на практике методикам CORA, NIST [1, 2], CRAMM, Digital Security Office, COBRA, RiskWatch [3], АванГард, Microsoft, ISO 17799:2000 [2], а также предложения по обобщению, изложенные в [4].

Исследование было разбито на несколько этапов:

- анализ требований стандартов и нормативных документов;
- формирование нового базового подхода на основе понятия риска несоответствия требованиям нормативных документов (см. [5]);
- обобщение подхода опросных листов с использованием полученных результатов;
- обобщение подхода табличных оценок и синхронизация обобщенного подхода с базовым.

Рассмотрим полученные результаты по каждому этапу.

Место исследования в предметной области

Поиск обобщающих характеристик подходов базового экспертного анализа информационных рисков в предметной области можно начать с оценки систем защиты информации (СЗИ). Согласно принятой в работе структуры оценки, отображенной на рис. 1, в качестве основы можно выделить экспертно-документальный метод проверки (исследование документации).

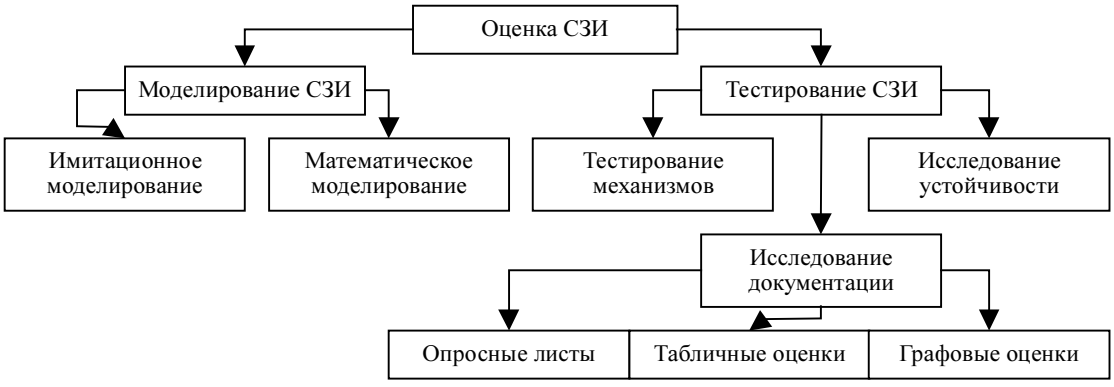

Рис. 1. Структура оценки СЗИ

[1] Работа поддержана грантом Президента молодым кандидатам наук № МК-3625.2007.9.

В данной статье не рассмотрены графовые оценки угроз и уязвимостей, поскольку находятся вне области обобщения.

Описание модели исследования

В работе рассматривается модель процедуры экспертной оценки системы защиты информации организации, которая представляет собой ориентированный взвешенный граф $G = (V, E)$, множество вершин описывает состояния системы с позиции экспертной оценки. Каждая вершина V_i ($i \in \{1, \ldots, N\}$, N – число вершин графа) описана следующими характеристиками: R_i, F_i, U_i, T_i, где R_i – интегральная качественная характеристика информационного риска; F_i – набор факторов, воздействующих на информацию; U_i – набор угроз и уязвимостей, активных в данном состоянии системы защиты информации; T_i – набор требований стандартов и нормативных документов, задействованных для данного состояния. Ребра графа описаны универсальными характеристиками переходов между состояниями системы – вероятность $P(T_s, T_l, T_o, T_t)$ невыполнения требований заданных подмножеств – соответственно стандартов, законодательных документов, организационной документации, технической документации. Таким образом, уникальными переходами между состояниями можно описать 16 состояний системы.

Каждое состояние характеризует порядок применения операций приведения в исходное (безопасное, то есть полностью соответствующее требованиям). Набор операций приведения уникален для каждого состояния и должен учитывать оптимальный путь приведения.

Рассмотрим такой набор операций конкретного состояния: $\alpha(V_i)$: A_f, A_u, A_t. Для каждого конкретного множества – A_f, A_u, A_t – операции задаются согласованно всеми экспертами, принимающими участие в оценке. При этом критична оценка адекватности выполняемых операций, например, по методам [4, 5].

Согласование экспертных оценок состояния системы

Задание видов операций приведения требует учета метода оценки каждого из элементов данных множеств. Согласно исследованию, выделены следующие базовые методы:

- опросных листов, позволяющих пользователям получить информацию из базы знаний экспертной системы в результате указания качественных характеристик процесса исследования;
- табличных оценок, позволяющих поэтапно оценить влияние факторов, воздействующих на ресурсы системы, на качественные характеристики процесса исследования;
- графоаналитические, позволяющие согласовать взаимное влияние угроз и уязвимостей, характеризующих различные состояния на графе G.

Кроме того, предусмотрен метод расчета финальных вероятностей перехода по графу G с использованием соотношений цепи Маркова.

Анализ требований стандартов и нормативных документов

Функционирование любой системы (организации) проходит, прежде всего, с учетом того правового поля, в котором она существует. В общем случае деятельность организации по обеспечению информационной безопасности осуществляется на основе следующих документов:

- действующих законодательных актов и нормативных документов РФ по обеспечению информационной безопасности;
- внутренних документов организации по обеспечению информационной безопасности.

Несоответствие документации организации каким-либо требованиям перечисленных документов приводит к появлению уязвимостей в системе, что может вызвать реализацию угрозы.

Можно проиллюстрировать необходимость анализа требований стандартов, используя метод стратификации и разделяя понятие идентификации риска:

- идентификация факторов, воздействующих на объект защиты;
- идентификация угроз и уязвимостей;
- идентификация требований стандартов и нормативных актов для формирования базы знаний экспертной системы.

На рис. 2 отображены переходы между упомянутыми уровнями, интерпретированные с использованием теоретико-множественного подхода.

Очевидно, при идентификации факторов, воздействующих на объект защиты, невозможно со сколько-нибудь приемлемой точностью оценить степень их влияния на конкретные элементы системы, но возможно проведение предварительной оценки. Множество факторов имеет отображение на множество угроз, которое в общем случае содержит большее количество элементов. Здесь необходимо ввести в анализ понятие качества изучения системы, которое влияет на степень детализации второго множества.

Отображая множество угроз и уязвимостей на множество требований стандартов и нормативных документов, можно увидеть, что такая операция делает анализируемое множество дискретным и четко определяет количество его элементов.

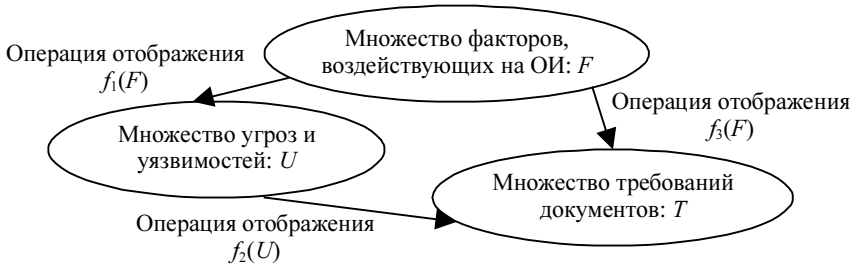

Рис. 2. Соотношение уровней идентификации информационного риска

Анализ требований стандартов и нормативных документов требует использования специальных методик формирования базы знаний экспертной системы, например указанной в [5].

Обобщение подхода опросных листов

Обобщенный подход опросных листов предполагает следующий метод извлечения информации из базы знаний экспертной системы. Используются качественные или количественные оценки состояния системы – вершины V графа состояний G системы, пользователь экспертной системы получает взвешенные рекомендации, разделенные согласно составляющих операции $\alpha(V)$. Анализ информационных рисков состоит из нескольких этапов (рис. 3):

- анализируются путем экспертной оценки с заданной статической либо динамической системой весов рекомендаций исходные данные по информационной системе;
- производится непосредственно анализ информационных рисков системы исходя из сформированных требований и исходных данных по организации, выдача практических рекомендаций по повышению уровня информационной безопасности и минимизации информационных рисков организации.

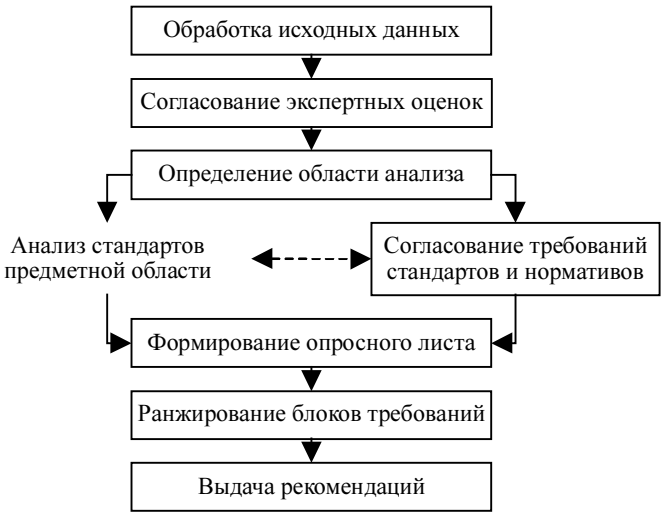

Рис. 3. Схема методики анализа остаточного информационного риска

Обобщенная методика отличается от классического понимания экспертного анализа рисков. На первом этапе классической методики происходит анализ всех возможных угроз и уязвимостей. Здесь же рассматривается только уровень требований нормативных документов, как упорядоченный и связанный непосредственно с предметной областью. Предполагается, что ранжирование угроз рассматривалось при составлении требований нормативных документов.

Анализ остаточных информационных рисков в данной работе основан на экспертных оценках, при этом учитывается несколько аспектов, позволяющих оценить их качество:

- квалификация эксперта;
- количество выданных рекомендаций в среднем по тематическим блокам;
- количество задействованных тематических блоков;
- средняя квалификация экспертов, выдавших конкретные рекомендации.

При этом алгоритм оценки принимает следующий вид (рис. 4):

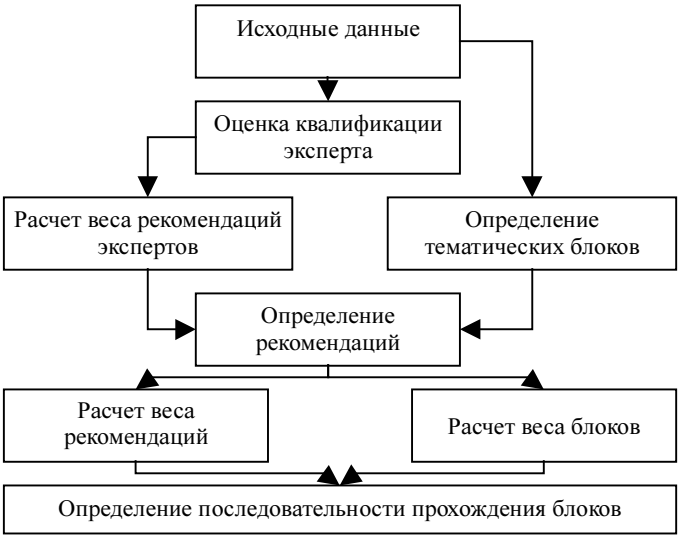

Рис. 4. Техника экспертного анализа

В данной методике существенны следующие моменты:

- эксперты, принимающие участие в оценке, могут иметь различную квалификацию;
- количество экспертов не ограничено сверху;
- рекомендации и вопросы опросного листа разбиваются по тематическим блокам, связанным с конкретными стандартами и нормативными актами;
- при опросе конкретного эксперта могут быть задействованы не все тематические блоки.

Преимуществом такой техники, по мнению авторов, является гибкость и независимость от конкретной ситуации, существенное достоинство – возможность добавления новых стандартов и нормативных документов. Тогда алгоритм опроса будет изменен путем введения новых тематических блоков и, возможно, удаления уже существующих.

Обобщение подхода табличных оценок

При таком подходе производится анализ факторов риска, т.е. источников угроз. При выборе факторов риска можно руководствоваться ГОСТ Р 51275-2006 «Объекты информатизации. Факторы, воздействующие на информацию». В данном случае можно анализировать как всю информационную систему организации в целом, так и ее отдельные составляющие. В общем виде анализ рисков, основанный на данном подходе, можно представить следующим образом (рис. 5):

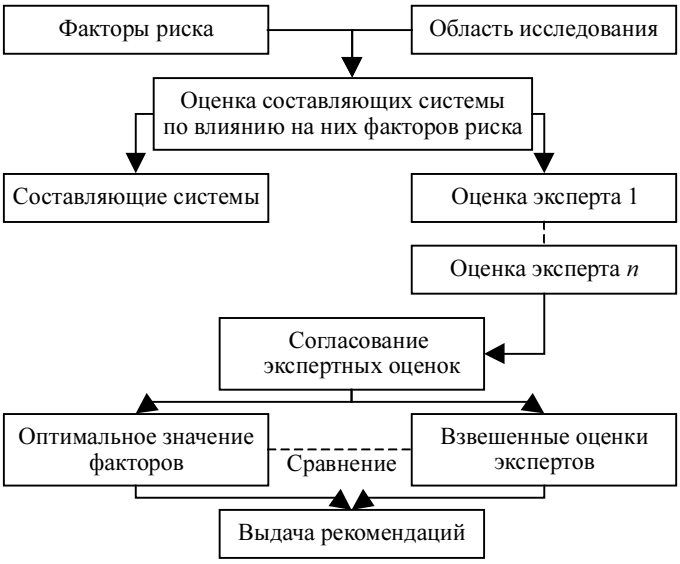

Рис. 5. Анализ факторов риска

Уровень каждого фактора оценивается экспертами, после чего проводится согласование экспертных оценок. Для удобства программной реализации необходимо получить шкалу оценки факторов риска – перевод качественных показателей в количественные – это осуществляется с помощью методов нечеткой логики. После получения количественных показателей можно определять функцию риска путем применения подходов математической логики.

Идея метода оптимизации следующая: по каждому фактору задается некоторое пороговое значение на шкале и производится оценка условно оптимального по заданному критерию значения фактора.

Выводы

Применение предложенных подходов позволяет упростить постановку задачи анализа информационных рисков автоматизированной системы. Использование экспертных систем в данных подходах является шагом к получению объективных результатов в процессе управления рисками.

Результатом работы является комплексный подход, формализующий требования к разработке некоторых видов экспертных систем заданной области.

ЛИТЕРАТУРА

1. *Прищеп С.В.* Сравнительный анализ методик NIST и CORA // Актуальные проблемы безопасности информационных технологий: Материалы I Междунар. науч.-практич. конф. Красноярск: Изд-во Сиб. гос. аэрокосмич. ун-та, 2007. С. 93 – 97
2. *Петренко С.А.*, *Симонов С.В.* Управление информационными рисками. Экономически оправданная безопасность. М.: Компания АйТи, ДМК Пресс, 2004. 384 с.
3. *Информационная* безопасность. М.: Компания АйТи, 2005. 468 с.
4. *Жданов О.Н.*, *Золотарев В.В.* Анализ информационных рисков при передаче данных по открытому каналу // Актуальные проблемы безопасности информационных технологий: Материалы I Междунар. науч.-практич. конф. Красноярск: Изд-во Сиб. гос. аэрокосмич. ун-та, 2007. С. 32 – 38.
5. *Золотарев В.В.*, *Ткаченко К.П.*, *Ширкова Е.А.* Управление информационными рисками несоответствия требованиям нормативных документов в области защищенного документооборота // Управление риском: науч.-технич. журн. Вып. 1. М.: Изд-во Анкил, 2008.

DOI 10.17223/20710410/2/16

УДК 681.322

РАСШИРЕНИЕ ФУНКЦИОНАЛЬНОСТИ СИСТЕМЫ БЕЗОПАСНОСТИ ЯДРА LINUX НА ОСНОВЕ ПОДМЕНЫ СИСТЕМНЫХ ВЫЗОВОВ

М.А. Качанов, Д.Н. Колегов

Томский государственный университет

E-mail: kden@sibmail.com

Обсуждаются некоторые известные средства безопасности для ядра Linux, основанные на перехвате и подмене системных вызовов, и предлагается модель системы реализации политик безопасности для ОС GNU/Linux на основе подмены системных вызовов, представляющая собой модуль ядра.

Ключевые слова: *перехват системных вызовов*, *политики безопасности*, *модуль ядра.*

В настоящее время существует значительное количество средств расширения систем безопасности и повышения уровня защищенности компьютерных систем (КС) на основе изменения функциональности ядер операционных систем (ОС) класса UNIX и Linux в частности. Большая их часть (SELinux, LSM, GrSecurity, Openwall, AppArmor и др.) устраняет уязвимости путем изменения исходного кода ядра ОС. Другие средства (Systrace, Janus, Ostia) используют перехват и подмену системных вызовов (СВ), но их реализация часто также затрагивает изменение кода ядра, библиотек или приложений и требует их перекомпиляцию.

СВ предоставляют интерфейс доступа к пространству ядра ОС из пространства пользователя для выполнения привилегированных операций. Основная идея заключается в перехвате СВ и выполнении функций, реализующих заданную политику безопасности (ПБ), вместо стандартных.

Так, на основе перехвата СВ возможно:

- реализовывать «песочницы» – среды, в которых действия процессов ограничены в соответствии с заданной ПБ [1, 2, 3];
- реализовывать различные формальные ПБ, например политику мандатного управления доступом Low-Watermark [4];
- обнаруживать и предотвращать атаки на уровне узла, осуществлять аудит в КС [3].

При этом имеется возможность не изменять исходный код ядра, а добавить требуемую функциональность применением загружаемых модулей ядра Linux. Такой подход используется, например, в реализации модели Low-watermark средства LOMAC. Недостатками данного подхода является сложность реализации, потенциальное увеличение количества уязвимостей в ядре и ограниченная переносимость. Целью данной работы является разработка и реализация загружаемого модуля ядра Linux, реализующего заданную ПБ на основе подмены СВ.

1. Обзор некоторых известных средств реализации ПБ

Исследования, связанные с подменой СВ, как одного из механизмов создания «песочниц», ведутся приблизительно с 1994 г. Классические системы такого рода, как правило, имеют *гибридную фильтрующую архитектуру* – на уровне ядра реализуются механизмы перехвата СВ и реализации ПБ, а на уровне пользователя – механизмы управления доступом [5, 6].

Основными проблемами систем с подобной архитектурой являются:

- неверное представление о состоянии ОС – для принятия решения в соответствии с ПБ необходимо учитывать возможные преобразования объектов;
- использование псевдонимов ресурсов;
- косвенные пути к ресурсам (UNIX domain sockets, core dumps, передача дескрипторов);
- состояние гонок;
- побочные эффекты запрещения системных вызовов.

Для устранения недостатков систем с фильтрующей архитектурой разработаны системы с *делегирующей архитектурой* (Ostia), где на уровне ядра запрещаются СВ, не соответствующие ПБ, а на пользовательском уровне агенты от имени ограниченных процессов осуществляют доступ к ресурсам [6].

Janus [1, 2, 3] – одно из первых средств реализации ограниченных сред выполнения недоверенного программного обеспечения (ПО) с помощью перехвата СВ. Первая версия Janus использовала механизмы виртуальной файловой системы /proc и системного вызова ptrace. Позднее было показано, что данные методы плохо подходят для изолирования процессов с точки зрения безопасности. Одним из недостатков реализа-

ции оригинальной системы Janus являлось отсутствие разрешения изменения корневого каталога для процесса [3].

В настоящее время Janus (J2) состоит из mod_janus – модуля ядра Linux, реализующего перехват СВ, и интерпретатора политик janus, выполняемого в пространстве пользователя. Механизмы обнаружения и предотвращения атак, автоматической генерации ПБ и аудита не реализованы.

Работа осуществляется по следующей схеме:

1) изолированный процесс выполняет СВ, который перехватывается;

2) модуль mod_janus уведомляет janus о наличии системного вызова, а процесс, выполнивший СВ, переводит в состояние сна;

3) janus запрашивает у mod_janus данные о процессе и принимает решение в соответствии с ПБ;

4) в зависимости от ПБ, системный вызов выполняется или возвращается сообщение об ошибке.

Systrace [3] является одним из самых известных средств создания ограниченных сред. Данное средство дополнительно позволяет обнаруживать и предотвращать атаки, осуществлять аудит и изменять (повышать) привилегии, что позволяет не использовать механизм setuid/setgid. На уровне ядра реализован быстрый механизм проверки системных вызовов на соответствие политике безопасности. При выполнении изолированным приложением СВ ядро выполняет его проверку без инспекции аргументов, при этом взаимодействия с препроцессором политик, расположенным в пространстве пользователя, не происходит. Обычно на этом уровне разрешаются такие системные вызовы, как read или write. Если ядро не может принять решение по системному вызову, то оно обращается к «демону» политик через устройство /dev/systrace, при этом системный вызов и его аргументы транслируются в слова системо-независимого языка.

Для файлов производится определение абсолютного имени и разрешение символических ссылок. При этом транслятор осуществляет нормализацию – оригинальные аргументы замещаются транслированными. Процесс на время принятия решения блокируется. Найдя политику для соответствующего вызова, «демон» политик возвращает в ядро соответствующий ответ, и процесс переходит в состояние готовности. Если политика не найдена, то решение запрашивается у пользователя через соответствующий интерфейс или СВ запрещается с возможностью возврата кода ошибок. Для схожих системных вызовов используется механизм псевдонимов для создания виртуальных СВ.

При аварийном завершении Systrace ядро завершает все приложения, находящиеся в состоянии мониторинга. Вновь созданные процессы наследуют политику процессов-родителей. Состояние гонок предотвращается описанным выше замещением аргументов СВ в ядре на аргументы, проверенные Systrace. Ядро выполняет только СВ, которые прошли проверку. Таким образом, можно, например, путем замещения аргументов, соответствующих именам файлов, организовать виртуальную файловую систему для приложения. Имеется возможность автоматической генерации ПБ. Для этого все выполняемые системные вызовы и их аргументы преобразуются в правила ПБ. Также реализована возможность повышения привилегий процессу для выполнения некоторых операций, например создания raw-сокета, благодаря чему приложения могут выполняться не с правами администратора системы. В настоящее время Systrace доступна для ОС NetBSD, OpenBSD и GNU/Linux.

Средство **LOMAC** [4] представляет реализацию модели мандатной ПБ Биба Low-Watermark в качестве загружаемого модуля ядра. После загрузки модуля происходит выделение сущностей высокого и низкого уровней целостности. К сущностям высокого уровня относятся службы ядра, системные файлы и приложения, библиотеки, а к низким – сервисы, к которым имеют доступ удаленные пользователи, скачанные файлы и т.д. Субъект получает доступ к объекту, если это не противоречит ПБ, реализуемым средствами ОС и LOMAC.

Ostia разработана Garfinkel [6] на основе анализа недостатков реализации Janus как делегирующий монитор ссылок, который осуществляет перенаправление СВ от изолированного приложения. Основными компонентами являются модуль ядра, библиотека эмуляции и агенты. Модуль ядра предназначен для запрещения выполнения всех СВ от изолированного ПО к защищаемым ресурсам напрямую. Когда модуль ядра перехватывает СВ, он передает управление обработчику, находящемуся в библиотеке эмуляции. Важно, что обработчик должен быть помещен в адресное пространство изолированной программы до ее выполнения и получения управления загрузчиком разделяемых библиотек – это достигается применением собственного загрузчика ELF-файлов. Затем происходит преобразование СВ в запрос к агенту. Запрос осуществляется через интерфейс сокетов. Каждый изолированный процесс имеет своего агента, который выполняет СВ в соответствии с принятой ПБ. В целях повышения производительности по умолчанию разрешенными считаются системные вызовы read, write, dup. Недостатком является необходимость исследования символических ссылок в пространстве пользователя для определения возможности доступа к файлу.

В [7] подмена СВ исследуется для возможности моделирования различных ПБ и реализации криптографических методов защиты информации в ОС. Развитием такого подхода является реализация ПБ в КС с помощью аспектно-ориентированного программирования (АОП). В [8] представляется инструмент, состоящий из языка АОП AspectTalk, виртуальной машины и транслятора с AspectTalk на язык этой машины, предназначенный для автоматизации процессов проектирования ПБ и их реализации в КС.

2. Описание схемы реализации ПБ

Расширение функциональности системы безопасности представляет собой реализацию политик безопасности, обеспечивающих контроль над выполнением системных вызовов. Каждая политика безопасности представляет собой однонаправленный связный список элементов отношения $S \times O \times A \times P$, где S, O, A и P – множества соответственно субъектов, объектов, субъектно-объектных взаимодействий и возможных действий. В качестве объектов могут выступать процессы и файлы; субъектами всегда являются процессы. Субъектно-объектными взаимодействиями могут быть отправка процессом сигнала другому процессу, чтение и запись в файл, порождение процесса и т.д. Возможными действиями являются разрешение, обход, запрет с возвратом кода ошибки, изменение самого взаимодействия (например, изменения содержимого буфера записи и посылаемого сигнала, комбинированные с протоколированием субъектно-объектного взаимодействия). В элементах политики безопасности субъекты идентифицируются по следующим параметрам: уникальный идентификатор, идентификатор пользователя, эффективный идентификатор пользователя, идентификатор группы, имя исполняемой команды, режим использования указанных параметров. Идентификация объектов происходит либо по вышеизложенному методу для процессов, либо по полным путям файлов в файловой системе для файлов. Субъектно-объектные взаимодействия определяются экземплярами политик безопасности – для каждого взаимодействия он свой. Действия задаются уникальным идентификатором, допустимым для данного субъектно-объектного взаимодействия, и кодом возможной ошибки. Под допустимостью понимается возможность применения действия к заданным субъекту, объекту и взаимодействию. Например, действие «удалить из буфера записи строку “string”» не может быть применено, если субъектом является процесс, объектом – файл, а взаимодействием – «открытие файла». Другими словами, допустимость действия определяется типами аргументов системного вызова и непосредственно самим системным вызовом (его номером).

Стоит уточнить, что каждому субъектно-объектному взаимодействию поставлено в соответствие некоторое подмножество множества всех системных вызовов (рассматриваются только те системные вызовы, разрешение структур ядра по аргументам которых, в принципе, позволяет определить некоторый файл или процесс). Данные подмножества не обязаны не пересекаться, но различным подмножествам не могут соответствовать одинаковые взаимодействия. При исполнении системного вызова идентификация текущего субъектно-объектного взаимодействия происходит путем поиска соответствий для множеств, которым принадлежит выполняемый в данный момент системный вызов. Если такое соответствие не единственно, то производятся анализ параметров, переданных системному вызову, и устранение такой неоднозначности. Например, пусть субъектно-объектному взаимодействию «создание файла» поставлено в соответствие множество системных вызовов {open, creat, mkdir}, а взаимодействию «открытие файла» соответствует множество {open}, и пусть выполняющийся системный вызов – open, и он не принадлежит никаким другим подмножествам, поставленным в соответствие другим субъектно-объектным взаимодействиям. Тогда указанный поиск соответствий даст в качестве результата два взаимодействия: «открытие файла» и «создание файла». В этом случае будет произведен анализ параметра flags (32-битный булев вектор) вызова open. Если побитовая конъюнкция вектора flags с наперед заданным вектором O_CREAT (24-битным), дополненным 8 нулевыми битами со стороны старших разрядов, даст ненулевой вектор, то результатом разрешения неоднозначности взаимодействия будет «создание файла», иначе – «открытие файла». Соответствие взаимодействий подмножествам системных вызовов задается так, что неоднозначность всегда можно устранить.

Дадим определения. *Активный субъект* – процесс, от имени которого выполняется системный вызов. *Активный объект* – объект, на который направлено действие системного вызова. *Соответствие* субъекта и объекта активным субъекту и объекту – совпадение характеристик первых с соответствующими характеристиками вторых с учетом режимов использования, указанных в элементе некоторой политики.

Контроль субъектно-объектных взаимодействий на основе политик безопасности осуществляется за счет перехвата системных вызовов следующим образом (см. рис. 1): в политике текущего взаимодействия осуществляется поиск элемента по субъекту и объекту, соответствующим текущим активным субъекту и объекту. Если такой элемент обнаружен, то выполняется действие, указанное в элементе политики; просмотр списка заканчивается, если только действие не является обходом, иначе выполняется действие по умолчанию. Таким образом, каждая политика является упорядоченной последовательностью фильтров для заданного взаимодействия.

На основе вышеизложенной схемы разработано средство, представляющее собой загружаемый модуль ядра Linux 2.6.x. Достоинствами такого подхода являются более высокая производительность по сравнению со средствами, использующими уровень пользователя, а также отсутствие необходимости изменения исходного кода ядра и его перекомпиляции и возможность динамической загрузки и выгрузки модуля в процессе работы операционной системы.

Политики безопасности представлены элементами структур ядра, имеющими поля, соответствующие характеристикам политик безопасности. Выбрано около 20 основных системных вызовов, удовлетворяющих оговоренным выше условиям.

Для каждого системного вызова написана функция-перехватчик с теми же аргументами, в действия которой входит:

1) определение текущего взаимодействия;

2) разрешение по переданным аргументам элементов структур ядра и абсолютных имен файлов с целью определения активных субъекта и объекта (разрешались дескрипторы, уникальные идентификаторы и относительные пути);

3) анализ соответствующего системному вызову экземпляра политики безопасности и выполнение вышеопределенного действия (рис. 1);

4) возврат кода ошибки, если это предусмотрено в элементе политики, либо возврат результата работы оригинального системного вызова.

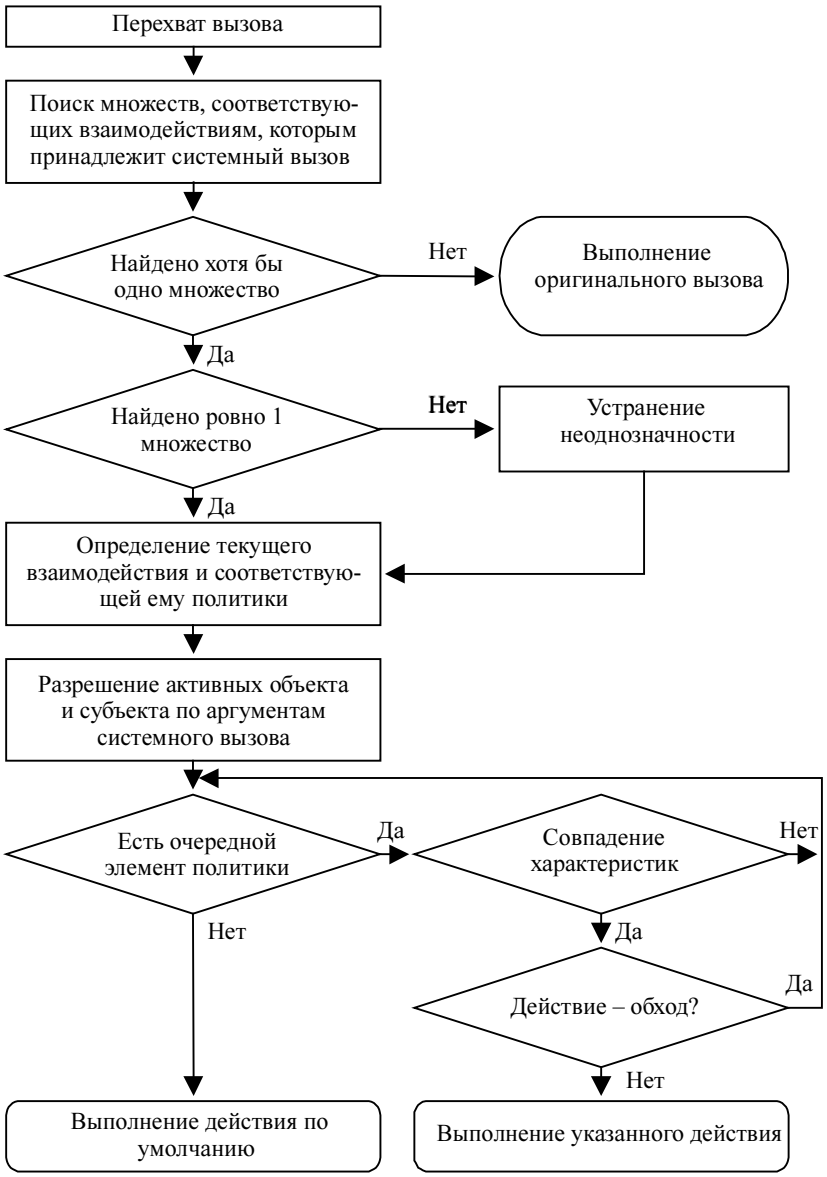

Рис. 1

3. Загружаемый модуль ядра, реализующий ПБ

При инициализации модуля вычисляется адрес таблицы системных вызовов, а адреса точек входа в ядро заменяются адресами соответствующих им функций-перехватчиков, при этом адреса оригинальных функций сохраняются в памяти. Таким образом, при всяком выполнении процессом системного вызова вызывается функция-перехватчик. Также в процессе инициализации модуля выполняются создание экземпляров политик и подготовка файла протоколирования, имя которого можно задать при загрузке модуля.

Для задания ПБ реализован интерпретатор, разбирающий строки ввода пользователя, написанные на определенном языке. Функциональными возможностями интерпретатора являются: добавление и удаление

элементов политик безопасности с указанием объектов, субъектов, взаимодействия и действия модуля, вывод политик безопасности и изменение порядка элементов в списке, т.е. управление фильтрацией. Причем вывод осуществляется на терминал, соответствующий текущему процессу, определяемому макросом current.

Взаимодействие с модулем ядра происходит по следующей схеме: интерпретатор разбирает строку ввода пользователя и в случае, если она корректна, формирует массив аргументов системного вызова execve, в котором в определенном формате указываются действия, выполняемые модулем ядра для формирования списка ПБ, и выполняет системный вызов. Модуль перехватывает системный вызов execve, разбирает строку аргументов и изменяет ПБ.

При выгрузке модуля происходит заполнение измененных элементов таблицы системных вызовов оригинальными значениями, выполняется освобождение памяти, выделенной под политики, и закрытие файла-протокола. Схема работы средства представлена на рис. 2.

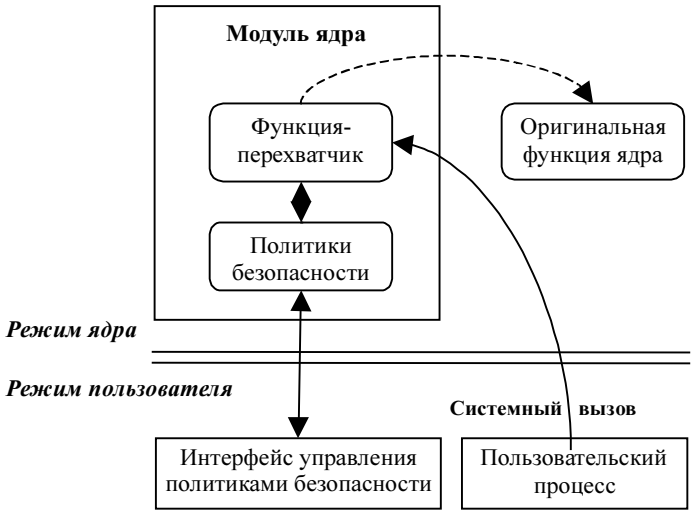

Рис. 2

В данной реализации с помощью модуля возможно осуществить сокрытие процессов и файлов, контроль порождения и клонирования процессов, контроль доступа к файлам, глубокий аудит выполняемых в компьютерной системе действий.

Заключение

В данной работе были описаны некоторые существующие средства расширения функциональности системы безопасности ядра Linux на основе подмены системных вызовов. Предложена собственная модель такой системы, использующая политики безопасности. Разработано средство, реализующее данную модель, в виде загружаемого модуля ядра с возможностью задания политик безопасности на определенном языке. Перспективными направлениями развития являются введение абстрактного типа ПБ на уровне реализации, исследование возможности реализации формальных моделей безопасности. Планируется также исследование возможностей применения данного средства и создание гибкого интерфейса работы со средством.

ЛИТЕРАТУРА

1. *Goldberg I*., *Wagner D*., *Thomas R*., *Eric A*. Brewer. A Secure Environment for Untrusted Helper Applications // Proceedings of the 6th Usenix Security Symposium, July 1996.
2. *Wagner D.A*. Janus: an approach for confinement of untrusted applications. Technical Report CSD-99-1056. 12. 1999. http://citeseer.ist.psu.edu/wagner99janus.html.
3. *Provos N*. Improving Host Security with System Call Policies, November 2002. http://citeseer.ist.psu.edu/provos02improving.html.
4. *Fraser T*. LOMAC: MAC You Can Live With. – USENIX, 2001. – http://portal.acm.org/citation.cfm?id=71575.
5. *Garfinkel T*. Traps and Pitfalls: Practical Problems in System Call Interposition Based Security Tools. www.stanford.edu/~talg/papers/traps/traps-ndss03.pdf.
6. *Garfinkel T*., *Pfaff B.*, *Rosenblum M*. Ostia: A Delegating Architecture for Secure System Call Interposition. http://citeseer.ist.psu.edu/garfinkel03ostia.html.
7. *Куликов М.Л.*, *Ромашкин Е.В.*, *Стефанцов Д.А.* Разработка средств моделирования политик безопасности операционных систем // Вестник ТГУ. Приложение. 2007. № 23. С. 189 – 193.
8. *Стефанцов Д.А.* Реализация политик безопасности в компьютерных системах с помощью аспектно-ориентированного программирования // Прикладная дискретная математика. 2008. № 1. С. 94 – 100.

2008 Математические основы компьютерной безопасности № 2(2)
DOI 10.17223/20710410/2/17
УДК 681.322

О СОРЕВНОВАНИЯХ CTF ПО КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

Д.Н. Колегов, Ю.Н. Чернушенко

Томский государственный университет

E-mail: kden@sibmail.com

Описывается концепция «CAPTURE THE FLAG» проведения соревнований по защите информации в компьютерных системах. Данный подход может быть использован для эффективного обучения по специальностям, связанным с обеспечением компьютерной безопасности.

Ключевые слова: *CTF*, *анализ защищенности*, *обучение*, *ролевые игры.*

В настоящее время известно несколько соревнований с концепцией «CAPTURE THE FLAG» (CTF) – DEFCON CTF, C.I.P.H.E.R, USF CTF и т.д. Далее речь пойдет о UCSB iCTF – международных распределенных соревнованиях по безопасности, организуемых Университетом Калифорнии Санта Барбары. Основная цель соревнований – проверка навыков проведения атак и защиты компьютерных систем (КС). В игре участвует неограниченное количество команд. Каждая команда имеет виртуальную сеть. В сети располагаются рабочие станции команды и защищаемый сервер, на котором запущено предоставляемое организаторами соревнований программное обеспечение, моделирующее уязвимую КС. Задача каждой команды – обеспечить состояние защищенности своих сервисов и атаковать сервисы других команд. Очки командам начисляются как за кражу флагов других команд путем компрометации сервисов, так и за защиту своих флагов. За доступностью сервисов и компрометацией флагов следит система подсчета очков (СПО).

Существует два основных вида CTF – CTF «с одной целью» и CTF «многие ко многим». В первом случае организаторами соревнований создается одна уязвимая система, а участники ее атакуют. Очки назначаются команде, сумевшей первой эксплуатировать уязвимость. Такой вид CTF часто организуется на конференции по безопасности DEFCON, которая сыграла значительную роль в популяризации игры. Во втором случае каждая команда и атакует, и защищается, имея в своем распоряжении копию уязвимой системы. Таким образом, выявление уязвимости позволяет команде защитить собственную КС и атаковать другие команды. Такой вид проведения CTF требует более сложную СПО.

CTF вида «многие ко многим» могут проводиться на уровне узла, когда все участники игры работают в одной системе, имея в своем распоряжении виртуальную (jail-подобную) систему, и на уровне сети, где все участники соединены по сети. У каждого из подходов имеются свои достоинства и недостатки. Достоинством CTF на уровне узла является то, что подсчет очков ведется путем простого чтения данных из специальных файлов различных виртуальных систем. К недостаткам следует отнести плохую масштабируемость и нереалистичность. CTF на уровне сети реалистичны, но для них более сложно обеспечить подсчет очков и соблюдение правил.

Турнир USCB CTF создан в 2003 г. сотрудником Университета Калифорнии Санта Барбары (UCSB) Джованни Вигна (Giovanni Vigna) на основе локальных соревнований DEFCON CTF. Основное отличие USCB CTF – распределенность и масштабность соревнований, а также неограниченное количество участников. Первоначально основные идеи игры использовались в процессе организации практических занятий по анализу защищенности компьютерных систем. В декабре 2003 г. были проведены первые соревнования среди 14 университетов США. В 2004 г. были проведены первые международные соревнования с участием команд из Норвегии, Италии, Германии, Австрии и США. Турнир 2005 г. проводился среди 22 команд, в том числе из Южной Америки и Австралии. С 2005 г. для обеспечения сетевого взаимодействия в процессе игры стали использоваться система Blending и GRE-туннели (до этого использовалась технология NAT), что значительно повысило реалистичность соревнований и упростило организацию.

В апреле 2008 г. на базе Уральского государственного университета им. А.М. Горького были проведены первые в России открытые межвузовские соревнования по защите информации с концепцией CTF – RuCTF-2008. Всего в игре приняло участие 9 команд из различных университетов России. Организаторами выступила команда Уральского государственного университета Hackerdom, имеющая большой опыт участия в соревнованиях подобного рода и занявшая 3-е место в UCSB CTF 2007.

Данная заметка написана по материалам из [1 – 3] и на основе собственного опыта участия авторов в UCSB CTF-2007 в составе команды Sibears Томского университета, впервые участвовавшей в подобных соревнованиях и занявшей в них 10-е место среди 35 команд университетов планеты.

1. Сетевая инфраструктура

Сеть USCB CTF (рис. 1) состоит из нескольких соединенных сайтов. Сайт включает одну или несколько сетей команд. Сайту выделяется множество немаршрутизируемых адресов класса B (например, 10.2.0.0/16), каждая команда в сайте получает множество немаршрутизируемых адресов класса C в соответствии с адресным пространством сайта (например, 10.2.1.0/24 и 10.2.2.0/24). В сети каждой команды размещены:

- узел team box (*tb*) с адресом x.y.z.1, обеспечивающий соединение с VPN-сервером игры main box (*mb*) через GRE-туннель;
- узел image box (*ib*, адрес x.y.z.2) с установленным ПО VMWare, предназначенный для эмуляции узлов vulnerable box (*vb*) и test box (*teb*), с адресами x.y.z.3 и x.y.z.4 соответственно. Узлы *tb* команд конфигурируются так, что любой исходящий трафик команд проходит через узел *mb*.

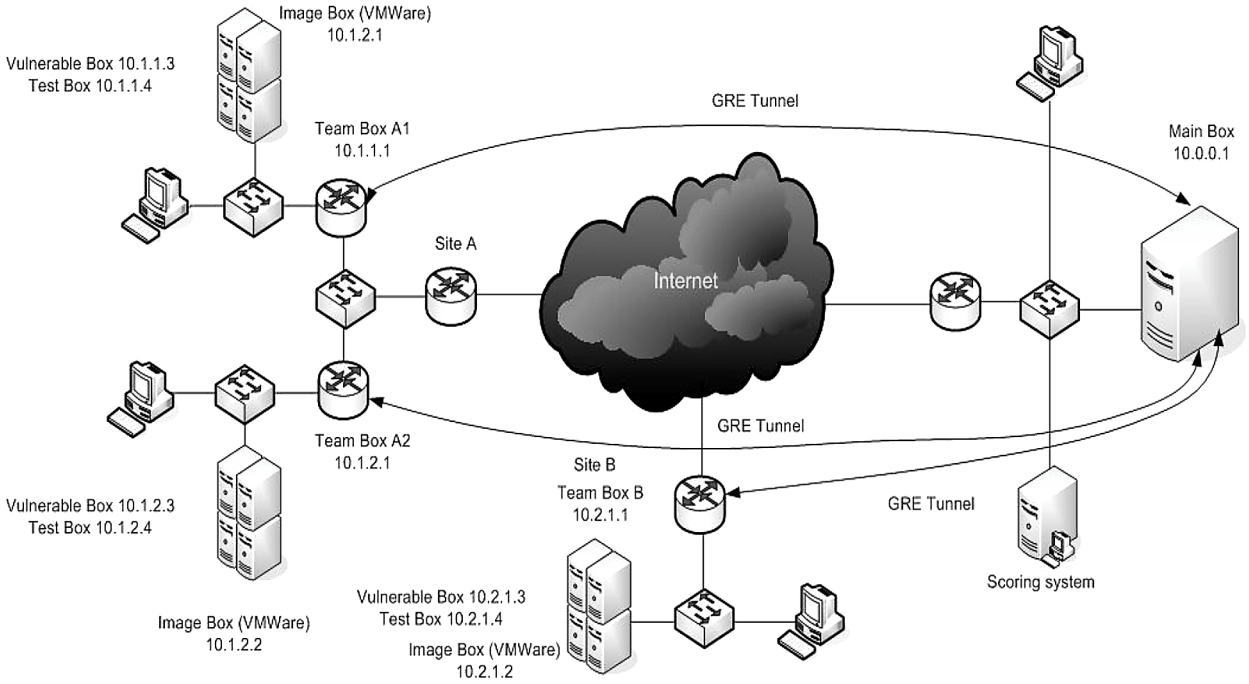

Рис. 1. Сетевая инфраструктура UCSB CTF

Узел *teb* предназначен для проверки готовности команды к соревнованиям, в процессе которой выполняются различные тесты, определяющие отсутствие фильтрации, прохождение дефрагментированного трафика, доступность сервисов, время прохождения датаграмм и т.д. Команды, не прошедшие проверку, к игре не допускаются до устранения несоответствий. Виртуальный узел *vb*, как правило, функционирует под управлением ОС GNU/Linux. Образ для *vb* создается организаторами соревнований и включает в себя несколько различных уязвимых сервисов (порядка 10), написанных на различных языках, и распространяется до начала игры в шифрованном виде.

2. Основные правила игры

В каждой команде назначаются faculty POC (point of contact) – лицо, ответственное за моральное поведение участников команды, и deployment POC – лицо, ответственное за функционирование сети и техническую подготовку команды. В соревнованиях участвуют только команды, имеющие отношение к образовательным учреждениям. Правила игры могут меняться, дополняться и уточняться. Основными правилами являются следующие:

- запрещены «массированные» DoS-атаки;
- запрещена генерация трафика DoS со взломанных узлов других команд;
- запрещены любые взаимодействия вне VPN;
- разрешены любые атаки на узлы VPN (в том числе и на рабочие станции);
- уязвимости должны быть устранены так, чтобы команды и система подсчета очков имели доступ к сервису, иначе очки с команды будут сняты;
- каждая команда должна поддерживать защищенность своих сервисов;
- продолжительность игры – 8 часов; первый час дается на подготовку, в течение которой атаковать другие команды запрещено.

3. Подсчет очков

Рассмотрим порядок действий СПО. В начальный момент времени t_0 СПО, используя служебную учетную запись, выставляет флаги для сервисов на узле *vb* каждой команды. Флаг – некоторый набор данных, ассоциированных с сервисом (например, `MTNzEwECA+hWjCb8t8/FgbEg/mSm1hbU231kjg=`). Затем в момент времени $t_1 = t_0 + q$ СПО пытается снова пройти регистрацию и получить доступ к сервису. Если регистрация проходит успешно и в момент времени t для $t_0 < t < t_1$ никто из команд не предъявил проверяемый флаг, то команде назначаются очки за защиту сервиса. Невозможность получить доступ к флагу расценивается как недоступность сервиса, и с команды списываются очки.

Считается, что команда A реализовала атаку в отношении некоторого сервиса команды B, если она может прочитать ассоциированный с этим сервисом флаг, сохраненный СПО, и затем предъявить его в заданный промежуток времени. Если флаг верный и еще не был обновлен СПО, то команде A начисляются очки за успешную атаку. Если несколько команд предъявили один и тот же флаг, то очки между ними делятся (простые уязвимости эксплуатировать легче, чем сложные). Временной интервал q свой для каждого сервиса.

Для предотвращения мошенничества в игре используется система *Blender*, задача которой – реализовать обращения СПО к сервисам от имени различных команд. Таким образом, значительно усложняется задача фильтрации входящего трафика с целью запрещения доступа к сервисам других команд и разрешения запросов от СПО. Общий принцип работы *Blender* заключается в сборе статистики о сетевом взаимодействии команд за некоторый период времени и затем генерации случайного раундового IP-адреса для доступа СПО к сервисам отдельной команды. Маскировка осуществляется с помощью технологии NAT на узле *mb*. Таким образом, СПО при проверке сервисов команды A с большей вероятностью будет использовать IP-адрес той команды B, которая наиболее активна в сетевом взаимодействии с A.

4. Организация игры

В команде должно быть не более 20 человек. Кроме оговоренных выше dpoc и fpoc, назначается или выбирается капитан команды. В процессе подготовки и проведения игры перед участниками стоят следующие практические задачи: администрирование КС, настройка межсетевых экранов и систем обнаружения и предотвращения атак, поиск уязвимостей в программном обеспечении (при наличии исходного кода и без него), сканирование сетей, написание и отладка эксплойтов и патчей, обнаружение вторжений, криптоанализ и др.

Сервисы, предоставляемые жюри в составе образа игры, пишутся на различных языках программирования – C, Java, Python, Ruby, PHP, Perl и др. Программное обеспечение может содержать ошибки, не позволяющие сервисам стартовать в системе или завершающие их через некоторое время.

Типовыми уязвимостями программного обеспечения являются SQL-injection, PHP-including, переполнение буфера, «гонки» и ошибки форматных строк. Встречаются также уязвимости конфигурирования КС – неправильные права доступа к каталогам, слабые и стандартные пароли.

Для каждой игры жюри придумывается легенда, объединяющая все задания и конкурсы (квесты) в одном тематическом плане, например: противостояние мафии и спецслужб, ограбление банка и т.д. В UCSB CTF-2007 для того, чтобы команда начала получать очки за защиту и атаку сервисов, ей было необходимо решить ряд головоломок различного уровня сложности. Пример такого задания – определение пароля, набранного злоумышленником, по видеофрагменту, записанному сотрудниками ФБР в результате наблюдения.

5. Организация обучения на основе CTF

На основе концепции CTF можно предложить пути для повышения эффективности подготовки специалистов по направлению анализа безопасности КС. Рассмотрим два варианта применения данной концепции. Первый из них заключается в проведении локальных CTF-соревнований со своей спецификой между командами обучающихся. Например, группа делится на 3 команды. Каждая команда по очереди выступает в качестве организаторов и жюри соревнований между двумя другими командами – придумывает замысел игры, создает образ игры, пишет уязвимые программы, осуществляет базовую настройку оборудования и т.д. Второй вариант состоит в предоставлении каждому обучающемуся своего образа CTF (локально или удаленно), на котором он сможет самостоятельно проводить анализ безопасности КС. Результаты выполнения заданий предоставляются организатору (преподавателю). Все образы CTF могут быть размещены на одном сервере при помощи современных технологий виртуализации КС. В последнем случае возникает непростая задача генерации образов CTF, так как все они должны быть в некотором смысле уникальными. Одним из решений может являться создание CTF-генератора, позволяющего генерировать множество различных образов из заданного шаблона. Данная задача в настоящее время исследуется автором USCB CTF Джованни Вигна.

ЛИТЕРАТУРА

1. http://www.cs.ucsb.edu/~vigna/CTF/
2. http://hackerdom.ru
3. http://ructf.org/

2008 Математические основы компьютерной безопасности № 2(2)
DOI 10.17223/20710410/2/18
УДК 681.3

О ЦЕНТРАЛИЗОВАННОМ ТЕРРИТОРИАЛЬНО-РАСПРЕДЕЛЁННОМ АНАЛИЗЕ СЕТЕВОГО ТРАФИКА

В.В. Лапшин

Томский государственный университет

E-mail: alter@mail.tsu.ru

Предлагается метод централизованного территориально-распределённого анализа сетевого трафика. Излагаются приёмы захвата, доставки, анализа и принятия решений. Сообщается о реализации метода и полученных экспериментальных результатах.

Ключевые слова: *сетевой трафик*, *захват трафика*, *анализ.*

С широким распространением Интернет возникли разные задачи, связанные с безопасностью сетей. Одной из этих задач является качественный анализ трафика по различным критериям оценок. В настоящее время качественный анализ с применением штатного аппаратного и программного обеспечения весьма затруднён. Главными причинами являются кратный рост пропускной способности сетей на базе Ethernet и архитектура ядер операционных систем, которая не приспособлена к транспортировке пакетов из пространства ядра в пространство пользовательского процесса при высокой скорости поступления пакетов. В данной работе предлагается способ использования захвата пакетов без потерь с интерфейса Ethernet с возможностью предоставления их прикладной программе, транспортировки в центр анализа из нескольких точек сбора трафика и анализа протоколов на основе полученных данных. В центре анализа трафика, создавая различные критерии анализа, можно строить распределённые системы выявления и блокирования DDoS-атак, несанкционированной сетевой активности, атак на сетевые службы, документирования сетевых сообщений, а также прочих задач информационной безопасности.

1. Проблема захвата IP-пакетов на скоростях 100 Мбит/с и выше

Современные операционные системы общего назначения (GNU/Linux, FreeBSD, Windows 2000/XP на Intel-совместимом аппаратном обеспечении) довольно легко справляются с захватом трафика на скоростях 10 Мбит/с и ниже через стандартный интерфейс библиотеки libpcap [1] (winpcap [2] для Windows). Количество необходимых операций для этого покрывается существующими мощностями аппаратного обеспечения. При захвате более высокоскоростного трафика наблюдается ситуация потери пакетов. Причиной этого является недостаточная вычислительная мощность ЭВМ, с одной стороны, и архитектура операционной системы – с другой, которая никогда не предполагала высокой (в пике до 1488000 переключений из контекста ядра в пространство пользователя и обратно при скорости 1 Гбит/с) интенсивности при передаче пакетов из пространства ядра в пространство пользователя.

Потеря пакетов при захвате чревата в первую очередь для процесса дальнейшего декодирования TCP/IP-сессий и других сеансовых соединений. В случае потери одного пакета процесс автоматизированного разбора протокола не может продолжить работу ввиду выпавшего необходимого участка данных. В этом случае вся последующая информация из данной сессии не может быть использована. Именно по этой причине недопустима потеря даже одного пакета.

С целью исправить ситуацию с большим количеством потерянных пакетов сторонними исследователями предложены разные варианты решения: libpcap-mmap [3], nCap [4] (исходные тексты в настоящее время не доступны, несмотря на некогда открытую лицензию), PF_RING [5]. Все они ориентированы на применение в рамках ядра Linux. Тем не менее ни одна из технологий не включена в стандартное ядро Linux. Организовав дистрибутив с интегрированной технологией PF_RING, можно добиться 100%-го захвата пакетов в 1Гбит/с-ethernet сети. Имея этот дистрибутив, можно быстро разворачивать точки захвата трафика. С его применением отпадает необходимость вручную накладывать заплатки и интегрировать технологию PF_RING в операционную систему.

2. О физических средствах захвата IP-трафика

Существует по крайне мере два подхода в методе захвата. Безболезненный метод заключается в прозрачном внедрении точки захвата трафика. Прозрачный метод позволяет сделать это без какого-либо изменения структуры сети, без установки дополнительных маршрутизаторов, без вмешательства в порядок работы пользователей. Реализовать это можно путём установки аппаратного зеркала ethernet-порта (такого, как nTap

[6], nMirror [7]) с подсоединением зеркальных выводов (для принимаемого и передающегося трафика, так как обычно ethernet-порты – полнодуплексные) к двум сетевым картам, где и будет производиться захват трафика. Под пиковой нагрузкой будем подразумевать ситуацию, когда канал загружен полностью и все захваченные пакеты должны быть проанализированы. Альтернативным механизмом может служить сетевой концентратор (hub). В этом случае достаточно будет одной сетевой карты для сбора принимаемого и передаваемого трафика.

При пиковой нагрузке к аппаратуре захвата должны быть подсоединены ещё две сетевые карты или одна, но более высокой производительности. Производительность последней должна быть не ниже удвоенной производительности менее скоростной. Таким образом, при использовании данной схемы удастся прозрачно и без нарушения работы сети произвести необходимые действия. Минусом метода является необходимость дополнительного подвода канала удвоенной ёмкости для передачи информации в центр анализа.

Второй метод может быть внедрён на базе маршрутизатора с операционной системой GNU/Linux. Работая в штатном режиме, ядро может быть дополнено функциональностью технологий PF_RING, после чего с произвольного порта может быть осуществлён захват пакетов. Недостаток метода состоит в том, что сетевые карты не должны быть произвольного производителя, так как применяемые технологии могут работать только с фиксированным числом сетевых карт, но обычно в маршрутизаторах на базе GNU/Linux установлены соответствующие карты от Intel или Broadcom (или иных подходящих марок). Кроме этого, минусом является необходимость обновления ядра после штатного системного обновления (так как оно с большой вероятностью заменит ядро и модифицированную библиотеку libpcap, и далее высокоскоростной захват пакетов без потерь будет невозможен), а также перерыв в работе при плановых перезагрузках сервера.

3. О необходимости централизованного пункта сбора и анализа трафика

Как показано выше, высокоскоростной трафик удаётся собрать в отдельных высокоскоростных участках сети. Но так как точек сбора может быть несколько, было бы не рационально в каждой устанавливать программное обеспечение по его анализу. Кроме этого, могли бы возникнуть многочисленные коллизии ввиду того, что один и тот же трафик может проходить через несколько точек сбора. Соответственно встаёт задача транспортировки захваченного трафика в центр сбора и его обработки. В целом можно отметить два пути: штатным образом через сеть Интернет, либо специализированным выделенным каналом, который одним концом подключен к точке сбора трафика, другим – напрямую к центру анализа трафика. В первом случае необходимо защитить при транспортировке трафик от третьих сторон, чтобы иметь уверенность в аутентичности и конфиденциальности при дальнейшем анализе.

Проверка вероятностного симметричного шифра libpssc на современном оборудовании показала, что он способен шифровать два потока по 100 Мбит/с. Libpssc представляет собой реализацию класса вероятностных поточных шифров, построенных на базе симметричных криптосхем в поточных режимах, описанных в работах [8, 9]. Использование других алгоритмов (AES, например) может предоставить возможность шифровать и более скоростной (1 Гбит/с) трафик. При транспортировке через выделенный канал защита может производиться как физическим, так и криптографическим методами. Таким образом, будем считать, что в описанных случаях трафик доставляется в центр анализа в полном объёме и недоступным для третьих лиц способом.

4. О возможном временном сохранении трафика

В случае временного перерыва связи с центром в пунктах сбора могут быть использованы накопители на жестких магнитных дисках для временного хранения информации (на криптографически-защищённых файловых системах или разделах). Криптографическая защита носителей предусматривается для случая возможной кражи или выемки несанкционированной стороной. После восстановления связи ранее записанная информация передаётся для анализа.

Протокол передачи представляет собой простой протокол, работающий через TCP-сессию. Соединение инициирует узел сборки трафика. Перед отправкой захваченных пакетов они подвергаются криптографическому преобразованию. В пункте приёма происходит обратная операция. Пункт приёма трафика по этому же каналу может передавать корректирующие сведения по поводу маски захвата трафика. Устанавливая маску, которая соответствует только интересующему трафику, можно снизить расходы ресурсов, связанных с транспортировкой трафика. Таким образом, в центр анализа будет доставлен только интересующий трафик.

В протоколе передачи предусмотрена возможность сжатия информации для снижения необходимого объёма полосы пропускания канала. В случае применения алгоритмов сжатия информации для текстовых и немультимедийных протоколов возможно экономить в среднем до 70 % пропускной способности канала.

5. Центр анализа и принятия решений

Предполагается, что центр анализа является защищённым от присутствия третьих лиц объектом. На его площадке ввиду отсутствия необходимости скрывать перехваченный трафик от третьих лиц происходит обратная процедура – расшифрование. Поток информации поступает на анализ произвольным средствам реа-

гирования, которые, руководствуясь заложенной в них логикой, принимают те или иные решения. Предполагается, что модули средств реагирования могут быть подключены (или отключены, заменены) в процессе работы системы через механизм подключаемых библиотек. Загрузка (отключение) одних модулей не влияет на работу других модулей. Одним из стандартных механизмов в этом процессе является блок сбора TCP/IP-сессий libnids [10] (основан на ядре Linux-2.0) из поступающих пакетов. Основной задачей данного механизма является получать данные TCP/IP-сессий из набора TCP-пакетов. В нём специально заложены средства учёта полученных дублируемых пакетов, возможных ввиду ненулевой вероятности прохождения одного и того же пакета через несколько точек сбора. Поэтому возникающие потоки являются уникальными и не дублируют друг друга. К особенностям механизма можно отнести возможность успешной дефрагментации поступающих пакетов, высокую надёжность и устойчивость против попыток злоумышленников блокировать сборку TCP/IP-сессий.

Получение TCP/IP-сессий не является достаточным для разбора произвольных протоколов. Далее могут использоваться специальные средства распознавания конкретных протоколов (HTTP, FTP, SMTP, POP3, ICQ и др.) и модули анализа трафика для должного реагирования на возникающие события.

6. Планы на будущее

Целью дальнейшего исследования является изучение методов оптимального построения математических алгоритмов распознавания и реагирования на (распределённые) DoS-атаки, проверка работоспособности метода на скоростях 10 Гбит/с и изучение вопроса об изменении производительности и масштабируемости при смене блока сбора TCP/IP-сессий на блоки из других ОС семейства BSD [11] и современной версии ядра Linux 2.6. Также не безынтересна перспектива использования в точках захвата ОС FreeBSD с разработкой аналога технологии PF_RING для сетевой подсистемы Netgraph, что поможет отказаться от поддержки специализированного дистрибутива на базе GNU/Linux и использовать стандартную ОС с подключаемым модулем ядра.

Заключение

В рамках апробирования предложенной схемы созданы специализированный дистрибутив на базе ОС GNU/Linux с интегрированной технологией PF_RING и программа приёма и передачи захваченного трафика в центр обработки. При транспортировке используются потоковые вероятностные криптосистемы, однако криптографические преобразования ими не ограничены. Программа приёма трафика с точек сбора производит обратную процедуру и предоставляет подключаемым средствам анализа необходимый интерфейс для обращения к нему. Таким образом, разработав соответствующие механизмы распознавания и реагирования и применяя предлагаемую систему в рамках ISP, возможно осуществить массовую защиту Web-сайтов от DDoS-атак, а также решать иные задачи компьютерной безопасности на множестве географически-распределённых точек присутствия узлов сбора трафика.

ЛИТЕРАТУРА

1. http://www.tcpdump.org/
2. http://www.winpcap.org/
3. http://public.lanl.gov/cpw/
4. http://www.nmon.net/nCap.html
5. http://www.ntop.org/PF_RING.html
5. http://www.nmon.net/nTap.html
6. http://www.nmon.net/nMirror.html
7. *Агибалов Г.П.* Вероятностные схемы симметричного поточного шифрования над конечным полем // Вестник ТГУ. Приложение. 2005. № 14. С. 39 – 42.
8. *Колегов Д.Н.* Общая схема вероятностной поточной шифрсистемы // Вестник ТГУ. Приложение. 2006. № 17. С. 112 – 114.
9. http://libnids.sourceforge.net/
10. http://ru.wikipedia.org/wiki/BSD

2008 Математические основы компьютерной безопасности № 2(2)

DOI 10.17223/20710410/2/19

УДК 004.056

ПРОБЛЕМА АУТЕНТИФИКАЦИИ В МНОГОУРОВНЕВЫХ ПРИЛОЖЕНИЯХ

П.А. Паутов

Томский государственный университет

E-mail: __Pavel__@mail.ru

В статье рассматриваются особенности аутентификации в многоуровневом приложении и возникающие при этом проблемы безопасности. Предлагается метод усиления защищенности наиболее распространенной в современных веб-приложениях двухзвенной схемы аутентификации.

Ключевые слова: *многоуровневые приложения*, *веб-приложения*, *безопасность веб-приложений*, *аутентификация в многоуровневом приложении*, *защита учетных данных для доступа к СУБД.*

Понятие многоуровневого приложения

Многоуровневое приложение (или *N*-уровневое приложение) – приложение, разделенное на *N* самостоятельных уровней, каждый из которых может выполняться на отдельной платформе. Типичным примером такого подхода является трехуровневая архитектура, которая подразумевает разделение приложения на следующие уровни:

1. Уровень представления – отвечает за представление данных для пользователя.
2. Уровень приложения (логики приложения) – отвечает за обработку данных в соответствии с логикой приложения.
3. Уровень данных – отвечает за хранение данных.

Трехуровневая архитектура используется в веб-приложениях. В веб-приложениях уровни распределены следующим образом:

1. Уровень представления – браузер.
2. Уровень приложения – программа, исполняемая на веб-сервере.
3. Уровень данных – СУБД.

Распределение приложения на несколько изолированных уровней позволяет достичь большей гибкости при масштабировании приложения или смене используемых технологий на каком-либо уровне [1].

Аутентификация в многоуровневом приложении

Применение трехуровневой архитектуры приводит к появлению нескольких звеньев аутентификации. В классическом двухуровневом приложении клиент аутентифицируется перед сервером. В трехуровневом приложении появляется второе звено аутентификации – клиент аутентифицируется перед прикладным уровнем, а прикладной уровень аутентифицируется перед СУБД. На рис. 1 представлен данный механизм с применением парольной аутентификации.

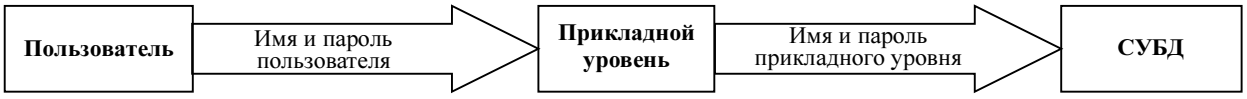

Рис. 1. Два звена аутентификации в веб-приложении

Далее можно выделить два частных случая:

1. Прикладной уровень имеет фиксированный набор пользователей СУБД, от имени которых он может работать.
2. Каждому пользователю приложения соответствует пользователь СУБД.

В современных веб-приложениях больше распространен первый подход. Вероятно, это обусловлено тем, что самые доступные тарифы хостинг-провайдеров ограничивают количество пользователей для доступа к СУБД (например, masterhost.ru, peterhost.ru). Чаще всего для взаимодействия с базой данных используется всего один пользователь. Например, приложения «1С-Битрикс» и «PHP-Nuke» используют данный метод [2, 3]. Таким образом, независимо от уровня привилегий пользователя приложения прикладной уровень всегда работает с СУБД от имени пользователя с максимальными привилегиями. Но современные веб-приложения, как правило, содержат несколько групп пользователей с различными привилегиями, например: «гость», «пользователь», «оператор», «администратор». Соответственно «администратор» обладает большими при-

вилегиями, чем «оператор», «оператор» имеет больше прав, чем «обычный пользователь», а «гость» обладает минимальным набором прав. При использовании для взаимодействия с СУБД только одного пользователя контроль прав доступа пользователей полностью осуществляется на прикладном уровне. Следовательно, если «обычный пользователь» как-то обойдет контроль прав на уровне приложения, то он получит доступ к СУБД с максимальными правами. Например, при успешной SQL-инъекции вставленный злоумышленником SQL-код будет выполнен с максимальными привилегиями. Реализация разделения прав также и на уровне СУБД позволяет уменьшить ущерб от таких атак. Для этого необходимо создать несколько пользователей СУБД, соответствующих группам пользователей веб-приложения, и назначить этим пользователям соответствующие привилегии [4]. То есть для доступа к базе данных будет использоваться один из нескольких пользователей, в зависимости от того, к какой группе пользователей относится пользователь приложения. При таком подходе, если, например, пользователь типа «оператор» как-то обойдет контроль доступа прикладного уровня, он сможет выполнить только те действия, которые позволены соответствующему пользователю СУБД. И, следовательно, ущерб от такой атаки будет меньше, так как привилегии пользователя СУБД «оператор» ограничены.

Защита аутентификационных данных для доступа к СУБД

В случае, когда для работы с СУБД прикладной уровень использует фиксированное количество учетных записей, возникает необходимость хранения аутентификационных данных этих записей (например, имен пользователей и паролей). Во многих современных веб-приложениях эти данные хранятся в открытом виде в некотором файле конфигурации на веб-сервере. Таким образом, при компрометации прикладного уровня злоумышленник может получить доступ к этим данным. Существующие подходы к решению данной проблемы в основном пытаются ограничить доступ к этим данным на уровне одного локального пользователя ОС сервера, на котором выполняется прикладной уровень [5 – 7]. Реквизиты для доступа к СУБД записываются в некоторый файл или хранилище, доступные только одному пользователю ОС, а сам веб-сервер запускается от имени этого пользователя. Такие методы не делают различия между пользователями веб-приложения, то есть независимо от того, какого типа пользователь («администратор» или «оператор») обращается к веб-приложению, прикладной уровень всегда будет иметь доступ к аутентификационным данным всех пользователей СУБД.

В данной работе предлагается защитить аутентификационные данные для доступа к СУБД, зашифровав их с помощью паролей пользователей веб-приложения. Такой подход позволяет не хранить реквизиты для доступа к СУБД в открытом виде, но при этом несколько усложняются операции по управлению пользователями. Далее рассматриваются три схемы реализации данного подхода. Для каждой схемы предлагаются алгоритмы выполнения следующих задач:

1. Смена пароля пользователем.
2. Создание нового пользователя.
3. Смена учетной записи СУБД.
4. Перевод пользователя из одной группы в другую.

1. Прямая схема. Для работы данной схемы прикладной уровень веб-приложения должен хранить таблицу, состоящую из двух колонок:

1) имя пользователя веб-приложения;

2) имя и пароль пользователя СУБД, соответствующего группе пользователя приложения, зашифрованные с помощью пароля пользователя веб-приложения.

Запись таблицы, соответствующая администратору приложения, должна содержать учетные данные всех пользователей СУБД. Данную таблицу можно хранить в БД. Для доступа к ней можно использовать пользователя СУБД, единственной привилегией которого был бы доступ к данной таблице на чтение. Такой пользователь СУБД может соответствовать группе пользователей приложения «Гости». Алгоритм аутентификации пользователя перед веб-приложением будет выглядеть следующим образом:

1. Пользователь веб-приложения передает свои имя и пароль прикладному уровню.
2. Прикладной уровень находит в описанной выше таблице имя пользователя и с помощью пароля пользователя расшифровывает учетные данные, необходимые для доступа к СУБД.
3. Прикладной уровень устанавливает соединение с СУБД от имени учетной записи, полученной на 2-м шаге.
4. В случае успешного соединения считается, что пользователь прошел аутентификацию.

Далее прикладной уровень работает с СУБД по установленному соединению от имени пользователя с привилегиями, соответствующими группе пользователя веб-приложения. Рассмотрим задачи по управлению пользователями.

1.1. Смена пароля пользователем.

1. Пользователь присылает свой старый и новый пароли.
2. Прикладной уровень расшифровывает учетные данные СУБД.

3. Прикладной уровень устанавливает соединение с СУБД для проверки правильности старого пароля.

4. В случае успеха шага 3 прикладной уровень зашифровывает учетные данные СУБД на новом пароле пользователя.

1.2. Создание нового пользователя. Создать нового пользователя приложения может только администратор.

1. Администратор присылает свой пароль и пароль нового пользователя.

2. С помощью пароля администратора расшифровываются учетные данные СУБД.

3. С помощью пароля пользователя зашифровываются учетные данные СУБД, соответствующие группе пользователя.

1.3. Смена учетной записи СУБД. При смене учетной записи СУБД необходимо зашифровать новые учетные данные на паролях соответствующих пользователей. В рамках данной схемы это требует получения паролей этих пользователей, что накладывает сильное ограничение на применение данной схемы.

1.4. Перевод пользователя из одной группы в другую. При переводе пользователя в другую группу необходимо также знать пароль пользователя, так как необходимо зашифровать реквизиты СУБД на пароле пользователя. Это ограничивает применимость данной схемы на практике.

2. Схема с мастер-ключом. В данной схеме вводится некоторый мастер-ключ, с помощью которого шифруются учетные данные пользователей СУБД. Таким образом, таблица из предыдущей схемы приобретает следующий вид:

1) имя пользователя веб-приложения;

2) мастер-ключ, зашифрованный с помощью пароля пользователя веб-приложения.

Необходимо также хранить таблицу с учетными данными СУБД, зашифрованными мастер-ключом. Рассмотрим задачи по управлению пользователями.

2.1. Смена пароля пользователем. Аналогично схеме 1.

2.2. Создание нового пользователя. Аналогично схеме 1.

2.3. Смена учетной записи СУБД.

1. С помощью пароля администратора расшифровывается мастер-ключ.

2. Новые учетные данные СУБД шифруются с помощью мастер-ключа.

2.4. Перевод пользователя из одной группы в другую. Перевод пользователя в другую группу не требует изменения данных в рассматриваемых таблицах.

Данная схема позволяет решить проблемы предыдущего варианта, но также имеет несколько серьезных недостатков:

1. При смене мастер-ключа мы сталкиваемся с теми же проблемами, что и в схеме 1 при смене учетных данных СУБД. Таким образом, мастер-ключ должен быть постоянным – такой подход может быть неприемлем для приложений, где требуется высокий уровень безопасности.

2. Если легальный пользователь системы сможет получить доступ к таблицам с учетными данными, то он сможет получить мастер-ключ и расшифровать все реквизиты СУБД. В итоге данный пользователь получит доступ к СУБД с максимальными привилегиями, тогда как в схеме 1 такой пользователь смог бы узнать только реквизиты СУБД, соответствующие его уровню доступа.

3. Схема с использованием асимметричного шифра. В данной схеме каждому пользователю системы ставится в соответствие пара – открытый и закрытый ключи. Таблица с учетными данными приобретает следующий вид:

1) имя пользователя веб-приложения;

2) имя и пароль пользователя СУБД, соответствующего группе пользователя приложения, зашифрованные с помощью открытого ключа пользователя;

3) открытый ключ пользователя;

4) закрытый ключ пользователя, зашифрованный с помощью пароля.

Такая схема позволяет решить проблемы двух предыдущих.

3.1. Смена пароля пользователем. Аналогично схеме 1.

3.2. Создание нового пользователя. Аналогично схеме 1, только необходимо сгенерировать открытый и закрытый ключи.

3.3. Смена учетной записи СУБД. При смене учетной записи СУБД необходимо с помощью открытых ключей соответствующих пользователей зашифровать новые учетные данные.

3.4. Перевод пользователя из одной группы в другую.

1. С помощью пароля администратора расшифровывается закрытый ключ администратора, а с помощью последнего расшифровываются учетные данные СУБД.

2. Учетные данные СУБД шифруются с помощью открытого ключа пользователя.

Данная схема позволяет успешно выполнять операции по управлению пользователями, а также обеспечивает более высокий уровень защиты, чем предыдущие схемы.

Заключение

Предложенные схемы позволяют защитить аутентификационные данные для доступа к СУБД и, таким образом, повышают уровень защищенности всего веб-приложения. Данный подход планируется реализовать в веб-приложении «Система тестирования знаний» [8] для экспериментальной проверки предложенного метода.

ЛИТЕРАТУРА

1. *Материалы* Свободной Энциклопедии. http://wikipedia.org.
2. *Karakas C.*, *Erba C.* PHP-Nuke: Management and Programming. 04.08.2005.
3. *Руководство* по инсталляции «1С-Битрикс: Управление сайтом 6.xx». 28.11.2007.
4. *OWASP*. A Guide to Building Secure Web Applications and Web Services 2.0 Black Hat Edition. July 27, 2005.
5. *Meier J.D.* Improving Web Application Security: Threats and Countermeasures. Microsoft Corporation, 2003.
6. *Newbigin J.* Keeping PHP Database Passwords Secure. 14.09.2005. http://uranus.it.swin.edu.au/~jn/linux/php/passwords.htm
7. *Secure* Database Access using PHP4, Apache, suexec, CGI, PostgreSQL, and peer sameuser. http://tril.tunes.org/admin/secure-db.html
8. *Паутов П.А.* Разработка и реализация веб-приложения «Система тестирования знаний» // Вестник ТГУ. Приложение. 2007. № 23. С. 194 – 197.

2008 Математические основы компьютерной безопасности № 2(2)
DOI 10.17223/20710410/2/20
УДК 004.056.52

УПРАВЛЕНИЕ ПРОЦЕССОМ ПРЕДОСТАВЛЕНИЯ ПРАВ ДОСТУПА НА ОСНОВЕ АНАЛИЗА БИЗНЕС-ПРОЦЕССОВ

Т.М. Пестунова, З.В. Родионова

Новосибирский государственный университет экономики и управления

E-mail: ptm@nsaem.ru, rodionova@nsaem.ru

В настоящей статье рассматриваются основные аспекты создания системы предоставления прав доступа пользователям и последующего управления изменениями на основе анализа бизнес-процессов организации.

Ключевые слова: *права доступа*, *бизнес-процесс*, *управление правами доступа*, *владелец процесса*, *eEPC*.

На сегодняшний день управление бизнес-процессами[1] организации – это один из самых распространенных путей значительного улучшения деятельности (сокращение затрат, повышение качества предоставления услуг и т.д.). Однако многие современные организации описывают и оптимизируют бизнес-процессы вне связи с их автоматизацией и защитой информации, в результате чего заложенные в автоматизированную информационную систему (АИС) сценарии зачастую не соответствуют бизнес-процессам и даже могут вступать в противоречие с ними, а система разграничения доступа (СРД) создает необоснованные помехи в работе пользователей.

На практике может возникнуть ситуация, когда бизнес-процессы предприятия организованы не самым оптимальным образом, вследствие чего через какую-то точку в структуре организации проходит информация, которая в данной точке для работы совершенно не нужна. Более того, в такой точке могут пересекаться потоки информации, к которой предъявляются существенно разные требования по защите.

Принимая во внимание, что одной из первых задач, решаемых при создании СРД, является категорирование корпоративной информации и определение прав доступа к ней различных сотрудников организации, очевидно, что АИС должна изначально строиться, исходя из требований проведения такого категорирования, а назначение прав доступа должно соответствовать организации бизнес-процессов предприятия.

На сегодняшний день достаточно распространенной практикой в управлении правами доступа является принятие решения о доступе к ресурсу руководителем подразделения (руководителем процесса) или владельцем процесса. Под владельцем процесса понимается должностное лицо, которое имеет в своем распоряжении персонал, инфраструктуру, среду, ведет мониторинг хода процесса и управление ходом процесса, несет ответственность за результат и эффективность процесса. В некоторых организациях функции руководителя подразделения и владельца процесса могут быть соединены и выполняться одним лицом (матричное конструирование процесса). Такой подход реализован в некоторых автоматизированных системах управления безопасностью: например таких, как «КУБ» (компания «Информзащита») и Identity Management (компания «Инфосистемы Джет»).

Подход на основе руководителя или владельца процесса имеет ряд недостатков, которые в целом связаны с доминированием дискреционного принципа назначения прав в такой модели. Помимо случайных ошибок, которые может допустить руководитель или владелец процесса, принимая решение о доступе, проблемы возникают и тогда, когда объекты используются на пересечении процессов. Два (и более) руководителя или владельца процессов должны принять решение о доступе, что создает почву для возникновения противоречий и дает возможность тому или иному владельцу процесса оказывать воздействие на работу стороннего процесса. Помимо всего прочего, владельцев процесса в организации на порядок больше, чем администраторов информационной безопасности, и, давая им такую власть, руководство увеличивает риск безопасности информации. С другой стороны, передать эти полномочия администратору информационной безопасности также не представляется возможным, так как эта категория сотрудников не обладает достаточными знаниями правил функционирования организации.

Управление процессом разграничения прав доступа на основе анализа бизнес-процессов позволяет практически исключить человеческий фактор и выйти на более формальный уровень принятия решения, основываясь на самой сути деятельности организации, ее бизнес-процессах.

[1] В современной практике управленческой и производственной деятельности для обозначения объектов моделирования принято использовать термин «бизнес-процесс». В МС ИСО 9000:2000 принят термин «процесс». Развитие и распространение двух областей знаний привело к сближению терминов. В дальнейшем термины «процесс» и «бизнес-процесс» будут использоваться как синонимы.

Рассмотрим в общем виде методику построения системы управления правами доступа и последующего управления изменениями на основе ролевой модели с использованием бизнес-процессов. Можно формализовать этапы создания системы управления правами доступа в следующем виде (рис. 1).

Рис. 1. Этапы создания системы управления правами доступа
на основе ролевой модели доступа с использованием бизнес-процессов

Первый этап создания системы управления правами доступа – это описание бизнес-процессов организации. Для проведения указанных работ необходимо определить способ описания. Не существует какого-то одного определенного способа описания, наилучшим образом отображающего деятельность организации, хотя «продвинутые» сотрудники многих организаций постоянно делают попытки его изобрести [1]. Наиболее часто для комплексного описания деятельности организации с целью внедрения автоматизированной системы используются следующие типы моделей:

- модели процессов управления (описание функций процесса, порядок их выполнения и управления, например в IDEF0, IDEF3, EPC);
- модель потока информации (например, в DFD);
- модель данных (например, в IDEF1X1).

Для выбора конкретной модели описания бизнес-процессов необходимо определить требования, предъявляемые к ней со стороны СРД.

Модель бизнес-процесса должна давать ответы на следующие вопросы:

- кто выполняет процедуры процесса (исполнитель);
- в каком из подразделений состоит тот или иной исполнитель;
- какие информационные ресурсы необходимо привлечь для выполнения процесса (например: информация о клиенте, информация о сделке и т.д.);
- какие функциональные модули участвуют в процессе;
- какие процедуры преобразования используются в функциональных модулях (например: копирование, удаление, чтение и т.д.);
- какие существуют связи между ролями и подсистемами.

Можно с уверенностью сказать, что модель данных не может использоваться в предлагаемой ситуации, так как не дает ответы ни на один ранее сформулированный вопрос. Модель потока информации также не рационально использовать, поскольку она не позволяет однозначно определить исполнителя процесса, подразделение, в котором он работает, а также процедуры преобразования. С помощью модели IDEF0 затруднительно вычленять информационные ресурсы, а определить, из какого подразделения исполнитель, вообще невозможно. Таким образом, остается модель eEPC (расширенная цепочка процесса), которая и дает ответы на большинство поставленных вопросов.

Общий вид модели бизнес-процесса в нотации eEPC представлен [2] на рис. 2.

Для прохождения следующих четырех этапов и заключительного построения модели доступа необходимо получить следующие данные из модели: множество пользователей системы, множество ролей пользователей, возможные домены ролей (подмножество объектов системы; например, систему можно разбить на отделы, так что каждый из начальников отделов будет играть роль «начальник» только в домене, соответствующем его отделу), множество информационных ресурсов, множество процедур преобразования, множество аналитических процедур, закрепление процедур за подсистемами.

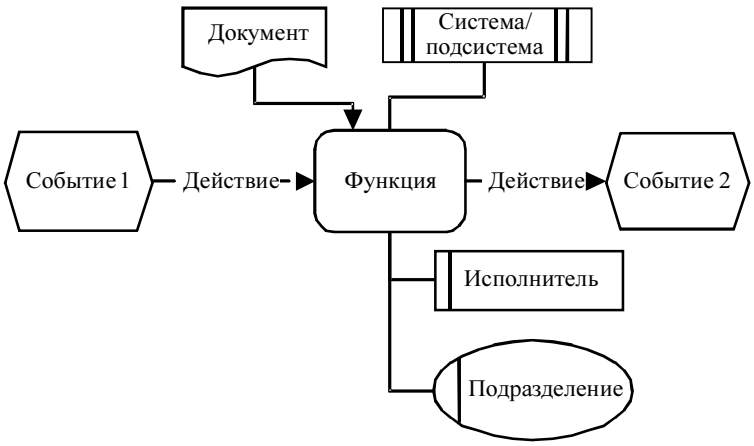

Рис. 2. Общий вид модели в нотации eEPC

Чтение модели позволяет получить следующие данные (табл. 1):

Таблица 1

Чтение модели

Графическое представление	Описание	Значение для СРД
Функция	Служит для описания функций (работ, процедур)	Категория информации, аналитические процедуры, информационные ресурсы
Событие 1	Служит для описания реальных состояний системы, управляющих выполнением функций	–
Действие	Описывает тип отношений между функциями	Процедуры преобразования; закрепление процедур преобразования за подсистемами
Документ	Отражает реальные носители информации, например бумажный документ	Частично категория информации, аналитические процедуры, информационные ресурсы
Система/ подсистема	Отражает прикладную систему	Для выявления принадлежности информационного ресурса к объектам автоматизации
Исполнитель	–	Данные для определения ролей
Подразделение	–	Данные для определения домена ролей

Таким образом, чтение модели eEPC позволяет получить всю необходимую информацию для построения модели управления правами пользователей.

После того, как модель построена и введена в действие, управление правами доступа осуществляется на основе сравнения состояния модели бизнес-процессов до и после внесения каких-либо изменений. Так, например, если создается новое подразделение, то в бизнес-процессе происходят соответствующие изменения, появляется новое подразделение, новые сотрудники, происходит перераспределение обязанностей. На основе анализа нового состояния модели бизнес-процесса в модель доступа добавляются новые роли, изменяются права доступа для уже существующих ролей и т.д.

Рассмотрим действие предложенной методологии на примере процесса «шифрования экзаменационных работ» при проведении вступительных испытаний в вузе. Под шифрованием письменных работ понимается присвоение каждой работе уникального кода, который обеспечивает невозможность идентифицировать автора. Процесс шифрования экзаменационных работ является достаточно значимым и защищаемым процессом вуза, он требует высокого уровня обеспечения безопасности шифрования. Автоматизация этого процесса реализована в АИС «Абитуриент», которая автоматизирует работу по приему документов, проведению вступительных испытаний и зачислению абитуриентов. Никто из участников процесса, кроме ответственного секретаря приемной комиссии, не должен получить информацию о том, кому принадлежит проверяемая работа, в этом выражается беспристрастность и объективность проверки. В общем виде процесс шифрования экзаменационных работ представлен на рис. 3.

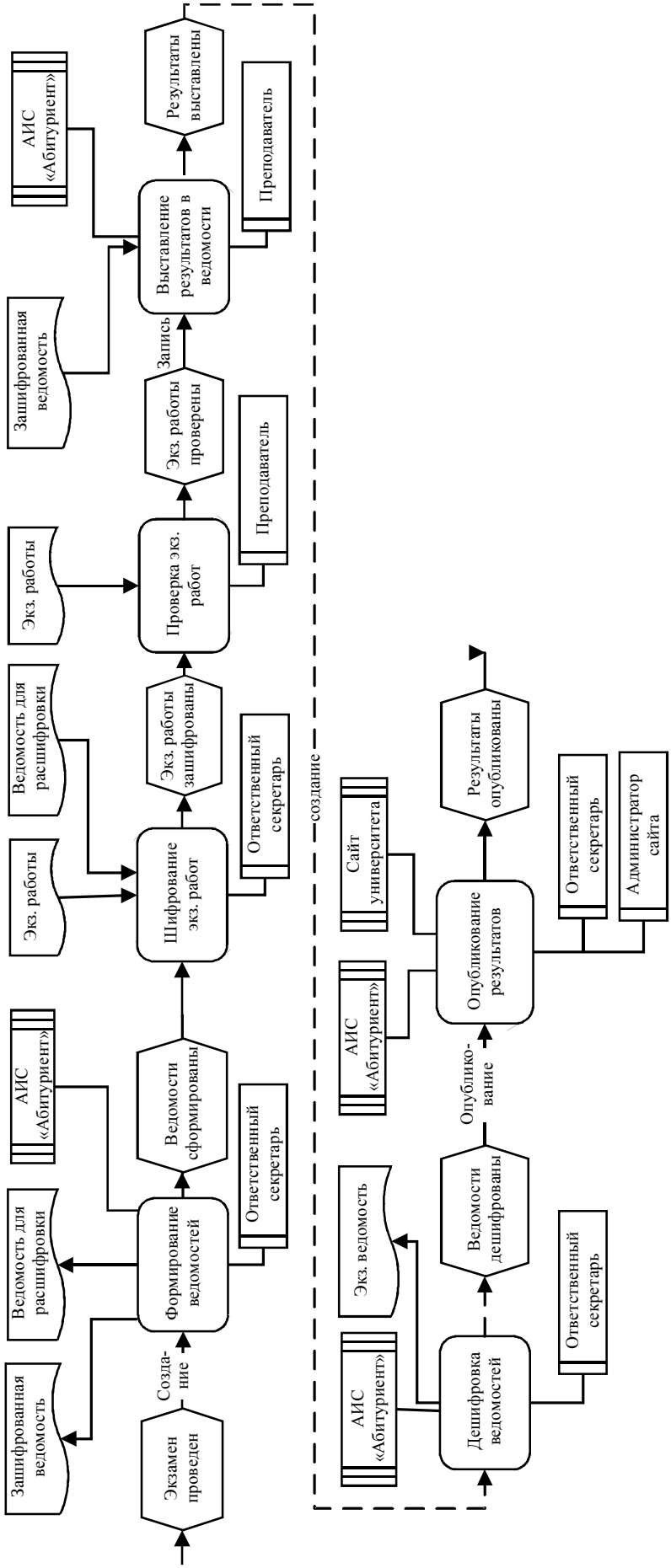

Рис. 3. Процесс шифрования экзаменационных работ при проведении вступительных испытаний в вузе

Второй этап (в соответствии со схемой рис. 1) по категорированию информации проводится на основе бизнес-процесса высшего уровня (формирование контингента вуза). При этом будем считать, что при категорировании вся используемая в рассматриваемом примере информация отнесена к одному классу, а все роли и объекты в примере связаны с функциональным модулем «Абитуриент».

Третий этап связан с выявлением ролей. Будем считать, что в вузе есть только одно подразделение, которое выполняет функции по приему абитуриентов – приемная комиссия, так что о выделении доменов ролей речь не пойдет. В качестве исходных данных для определения ролей можно идентифицировать всех «исполнителей» процесса: ответственный секретарь, преподаватель, администратор сайта. Конечно, не всегда пользователи информационной системы определяются исходя из организационно-штатной структуры, это может происходить и на основе похожести индивидуальных потребностей пользователя, но рассмотрение этого подхода выходит за рамки целей статьи. В указанном примере вполне применим подход выделения ролей исходя из организационно-штатной структуры.

Следующий этап связан с выявлением информационных ресурсов, которые можно выделить из функций (только те функции, которые выполняются с помощью АИС) и документов (учитываются только те бумажные документы, которые изначально формируются в АИС и являются объектами системы): формирование ведомости; результаты экзаменационных работ; зашифрованная ведомость; ведомость для расшифровки; ведомость.

Процедуры преобразования отображаются на диаграмме с помощью объекта «действие»: создание; запись; опубликование.

Все вышеуказанные действия позволяют сформировать простейшую модель доступа (табл. 2):

Таблица 2

Распределение доступа (подсистема «Абитуриент»)

Роль	Процедура преобразования	Информационный ресурс
Ответственный секретарь	Создание	Формирование ведомости
	Запись Опубликование	Результаты экзаменационных работ
	Создание	Зашифрованная ведомость
	Создание	Ведомость для расшифровки
	Создание	Ведомость
Преподаватель	Запись	Результаты экзаменационных работ (зашифрованных)
Администратор сайта	Опубликование	Результаты экзаменационных работ

ЛИТЕРАТУРА

1. *Репин В.В.*, *Елиферов В.Г.* Процессный подход к управлению. Моделирование бизнес-процессов. М.: СИА «Стандарты и качество», 2004. 210 с.
2. *Каменнова М.*, *Громов А.*, *Ферапонтов М.*, *Шматалюк А.* Моделирование бизнеса. Методология ARIS. Практическое руководство. М.: Весть-Метатехнология, 2001. 327 с.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ НАДЕЖНОСТИ ВЫЧИСЛИТЕЛЬНЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

DOI 10.17223/20710410/2/21

УДК 681.3.012+681.3-192

ЭФФЕКТИВНОСТЬ ЛОКАЛЬНОГО САМОДИАГНОСТИРОВАНИЯ В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С ЦИРКУЛЯНТНОЙ ДИАГНОСТИЧЕСКОЙ СТРУКТУРОЙ[1]

Ю.К. Димитриев

Институт физики полупроводников СО РАН, г. Новосибирск

E-mail: dimi@isp.nsc.ru

Рассматривается системное диагностирование в присутствии неисправностей с кратностью не более t. Анализируются условия, при которых состояние каждого модуля системы определяется только по исходам тестирования модулей, имеющих с ним физические связи (локальная диагностируемость). Показана эффективность локальной t-диагностируемости системы при использовании избыточности по числу исходов тестирования, участвующих в сопоставительном анализе.

Ключевые слова: *мультипроцессорные вычислительные системы*, *системная локальная диагностика*, *теоретико-графовая модель.*

В данной работе продолжается начатое работами [1, 2] исследование возможности диагностирования мультипроцессорных вычислительных систем (ВС) при отказах кратности не более t. Используется теоретико-графовая модель системы, предложенная в работе [3], Препараты – Метца – Чжена (ПМЧ-модель). Диагностическая структура ВС представлена циркулянтным графом, имеющим вершинную степень t и число вершин N. Такой граф обеспечивает определение состояния ВС при неисправностях, кратность которых не превышает значения t (t-диагностируемая ВС), при $N \geq 2t + 1$ по результатам сопоставительного анализа исходов тестирования всех модулей системы (глобальное диагностирование).

В работе [1] предложены правила, с помощью которых определение технического состояния каждого модуля осуществляется по результатам тестирования только тех модулей, которые имеют с ним физические связи. Эти правила названы *правилами самоопределения*, а базирующаяся на их использовании диагностика – *локальной*. Использование локальных правил самоопределения позволяет разрабатывать распределенные методы диагностирования.

Показано, что для заданного t существуют границы по числу модулей в системе, представленной циркулянтным графом, вне которых локальные правила обеспечивают самоопределение не для всех модулей системы. Подобные состояния ВС называем тупиковыми, а диагностирование – частичным.

В работе [1] показано, что локальная t-диагностируемость в случае тупиковых состояний достигается введением избыточности в числе тестов, выполняемых над системой. Однако тестирование занимает основную часть времени диагностирования, так что применение дополнительных тестов уменьшает время использования ВС по назначению и реактивность ВС на возникшую неисправность. Для уменьшения времени диагностирования в работе [2] предложено использовать избыточность в числе исходов тестирования, участвующих в сопоставительном анализе с целью достичь самоопределения состояния модулей. Для этого используются исходы тестирования, соотнесенные со структурными единицами циркулянтного графа, так называемыми треугольниками тестирования.

В данной работе изучена эффективность самоопределения модулей при использовании таких треугольников тестирования.

1. Обозначения и определения

Диагностической моделью вычислительной системы, в которой допустимы множественные отказы и ненадежные тесты, является ориентированный граф $D = (V, E)$. Вершины $i \in V$ графа сопоставлены модулям, а дуги $(i, j) \in E$ – тестовым связям между ними; $(i, j) \in E$, если модуль, сопоставленный с вершиной i, может проверить и оценить состояние модуля, сопоставленного с вершиной j. Вес $a(i, j)$ дуги (i, j) диагностического графа отображает оценку состояния проверяемого модуля j, полученную проверяющим модулем i после

[1] Работа выполнена при поддержке Российского фонда фундаментальных исследований (грант 05-08-01301-а).

выполнения им соответствующего теста. Вес $a(i, j) = 0$, если модуль i считает исправным модуль j, иначе $a(i, j) = 1$. Согласно используемой ПМЧ-модели, исход теста, выполняемого неисправным модулем, не зависит от фактического состояния тестируемого модуля.

Множество F_k, $F_k \subseteq V$, неисправных модулей, одновременно присутствующих в системе, называем образом неисправностей. Для всякого F_k справедливо $|F_k| \le t$, где $|X|$ обозначает мощность множества X. Множество допустимых неисправностей системы составляют все возможные сочетания из N модулей по n, где n пробегает значения от 1 до t. Таким образом, множество $\{F_k\}$ допустимых состояний ВС зависит от значений t и N и обозначается $F(t, N)$.

Упорядоченную совокупность оценок, получаемую при выполнении всех возможных тестов над системой в присутствии F_k, называем синдромом, порождаемым F_k, и обозначаем $\sigma(F_k)$; $\sigma(F_k) = \{a(i, j): (i, j) \in E\}$.

Диагностическая структура ВС представлена t-диагностируемым циркулянтным графом ($D_{1,t}$ (N)-граф) с параметрическим описанием (N; 1, 2, …, n, …, t), где $N = |V|$, а $n = 1, 2, \ldots, t$ – прыжки. В циркулянте $D_{1,t}$ (N) вершины перенумерованы и каждая i-я вершина ($0 \le i \le N - 1$) смежна с вершинами $(i \pm 1) \bmod N$, $(i \pm 2) \bmod N$, …, $(i \pm t) \bmod N$, а дуга $(i, (i + n))$ имеет метку соответствующего прыжка $s_n = n$.

Пусть в диагностическом графе множества $X(i)$, $Y(i)$ и $H(i)$ представляют соответственно модули тестирующие, тестируемые и смежные для модуля, сопоставленного вершине i; $X(i) = \{j \in V: (j, i) \in E\}$, $Y(i) = \{j \in V: (i, j) \in E\}$, $H(i) = X(i) \cup Y(i)$ и $X(i) \cap Y(i) = \varnothing$. Согласно весу дуг, инцидентных вершине i, имеем $X(i) = X_0(i) \cup X_1(i)$, $Y(i) = Y_0(i) \cup Y_1(i)$. При этом $X_0(i) \cap X_1(i) = \varnothing$, $Y_0(i) \cap Y_1(i) = \varnothing$.

Для упрощения изложения распространим терминологию ВС на представляющий ее граф и в дальнейшем используем выражения «состояние графа» вместо «состояние системы», «исправная (неисправная) вершина» вместо «исправный (неисправный) модуль», «исход теста (i, j)» вместо «вес дуги (i, j)» и т.п.

2. Условия самоопределения вершины

Пусть заданы $D_{1,t}$ (N)-граф, F_k и $\sigma(F_k)$. Каждой вершине i графа сопоставим метку $m(i) \in \{\alpha, 0, 1\}$. Если по результатам анализа $\sigma(F_k)$ состояние вершины i не идентифицировано, то $m(i) = \alpha$, если вершина i признана исправной или неисправной, то $m(i) = 0$ или $m(i) = 1$ соответственно (такие состояния называем финальными, а сами вершины – самоопределимыми). Совокупность меток вершин, равно как и процесс их определения, называем разметкой графа.

Разметка графа осуществляется по шагам. Если на очередном шаге идентифицировано финальное состояние вершины i, то в дальнейшем возможность самоопределения для вершин $j \in H(i)$ рассматривается на остаточном множестве $R(j)$, образованном исключением из $H(j)$ вершины i (равно как и других смежных вершин в финальном состоянии).

Динамика изменения состояния графа в процессе диагностирования характеризуется остаточным образом неисправностей $F_k - f$, где $f = \cup_{i \in V} f(i)$ и $f(i)$ – множество смежных с i неисправных вершин, идентифицированных на предшествующих шагах. Каждой вершине i графа сопоставлены значение $|F_k - f(i)|$ порога самоопределения и $\tau(i)$ величины корректировки этого порога.

Начальное значение порога самоопределения для всех вершин равно величине кратности неисправностей t, поскольку априорная информация о мощности образа неисправностей отсутствует. Порог меняется по мере установления финального состояния для новых вершин.

Состояние системы вычисляется в результате проверки правил самоопределения, рекурсивно выполняемой для каждой вершины графа:

$$[|(X_1(i)) \cup (Y_1(i))| > (t - \tau(i))] \mapsto m(i) := 1; \tag{1}$$

$$[|(X_0(i)| \ge (t - \tau(i))] \mapsto m(i) := 0; \tag{2}$$

$$m(i) := 0 \mapsto [\forall j \in Y_0(i)\{m(j) := 0\}] \,\&\, [\forall j \in (X_1(i) \cup Y_1(i))\{m(j) := 1\}]; \tag{3}$$

$$m(i) := 1 \mapsto \forall j \in X_0(i)\{m(j) := 1\}. \tag{4}$$

Здесь $m(i) := c$ обозначает операцию присвоения вершине i метки финального состояния: $c \in \{0, 1\}$. Выражения (1) – (2) указывают признаки финального состояния вершины i, исходя из значений синдрома, относящихся только к ней самой. Выражения (3) – (4) представляют условия установки финального состояния смежных с i вершин, являющейся следствием идентификации финального состояния вершины i согласно (1) и (2). Установка финального состояния вершин согласно выражениям (3) – (4) следует из правил определения исходов тестирования для ПМЧ-модели, в соответствии с которыми $[Y_0(i) \subset (V - F_k) \,\&\, [X_1(i) \cup Y_1(i)] \subseteq F_k]$, если вершина i исправна, и $X_0(i) \subset F_k$, если вершина i неисправна.

Пусть для вершины i диагностического циркулянта $\{j, k\} \subset H(i)$. Рассмотрим ориентированный полный граф K_3 из вершин $\{i, j, k\}$. Назовем треугольником тестирования ориентированный подграф графа K_3, не содержащий взаимно обратных дуг. Для вершины i диагностического циркулянта множество треугольников тестирования на множестве $\{i, j, k\}$ задается выбором дуг K_3. В работе [2] выделены два базовых треугольника тестирования, описываемых наборами дуг: $A_1 = \{(i, j), (i, k), (j, k)\}$ и $A_2 = \{(i, j), (j, k), (k, i)\}$. Показано, что все остальные треугольники тестирования эквивалентны базовым с точностью до обозначения вершин. В дальнейшем описанные треугольники тестирования обозначаем перечислением вершин (i, j, k), имея в виду указанную выше последовательность его ребер.

В работе [2] показано, что для треугольника тестирования A_1 синдромы $\{a(i, j), a(i, k), a(j, k)\} = 001$ или $\{a(i, j), a(i, k), a(j, k)\} = 010$ соответствуют неисправной вершине i при любом состоянии вершин j и k. В треугольнике тестирования вида A_2 синдром $\{a(i, j), a(j, k), a(k, i)\} = 001$ соответствует неисправной вершине i, синдром $\{a(i, j), a(j, k), a(k, i)\} = 010$ – неисправной вершине j, а синдром $\{a(i, j), a(j, k), a(k, i)\} = 100$ – неисправной вершине k. Вершину, для которой по синдрому, сопоставленному треугольнику тестирования, можно указать финальное состояние, называем самоопределимой, а сам треугольник тестирования – определяющим.

Из всего множества треугольников тестирования, которые можно построить на множестве $V(i) = i \cup H(i)$, для нас представляют интерес те, которые обеспечивают самоопределение вершины i. Множество этих треугольников соответствует случаю, когда диагностирование обеспечивается при минимуме анализируемой информации для каждой вершины. Для вершины $j \in H(i)$ идентификация финального состояния обеспечивается из аналогичных треугольников тестирования, построенных на множестве $j \cup H(j)$. Поэтому сказанное выше резюмируем в виде следующих формальных отношений:

$$\{a(i, j), a(i, k), a(j, k)\} = \{001, 010\} \mapsto m(i) := 1; \tag{5}$$

$$\{a(i, j), a(i, k), a(j, k)\} = \{100\} \mapsto m(i) := 1. \tag{6}$$

Треугольник тестирования вида A_1 строится на вершинах из $i \cup Y(i)$ и может использоваться для самоопределения вершин при любых значениях N. Область использования треугольника вида A_2 ограничена значениями $2t + 1 \le N \le 3t$, поскольку он образуется на множестве $i \cup X(i) \cup Y(i)$.

Определение 1. Остаточный образ неисправностей $F_k - f$ называем *самоопределимым*, если и только если при любом совместном с ним синдроме хотя бы для одной вершины из $V - f$ выполняются условия самоопределения (1) – (6). В противном случае образ называется *тупиковым*.

Определение 2. Граф называем *локально t-диагностируемым*, если и только если для каждого $F_k \in F(t)$ любой из его остаточных образов неисправностей при $F_k - f \neq \varnothing$ самоопределим при всяком $\sigma(F_k)$.

При заданной кратности неисправностей t область значений N, в которой возможно возникновение тупиковых состояний, ограничена значениями $N = 2t + 1$ и некоторым эмпирически определяемым значением $N_b > N$. N_b представляет собой наибольшее число вершин циркулянта, при котором еще существуют такие образы неисправностей мощности t, что для всякой исправной вершины найдется хотя бы одна тестирующая ее неисправная вершина. Описанное свойство называем свойством топологического покрытия графа.

Будем говорить, что неисправная вершина j 1-покрывает (0-покрывает) вершину i, если $a(j,i) = 1$ ($a(j,i) = 0$). Образы неисправностей, топологически покрывающие граф, могут порождать синдромы, при которых для каждой исправной вершины i найдется 1-покрывающая неисправная вершина j (условие тупикового состояния для исправной вершины), а для каждой неисправной вершины число инцидентных дуг с весом 1 не превышает текущего порогового значения (условие тупикового состояния для неисправной вершины). В работе [1] значение N_b характеризуется как такое наибольшее число вершин циркулянта степени t, при котором

$$\max_{F_k \in F(N_b, t)} \{\sum_{j \in F_k} \Delta(j)\} \ge N_b - t. \tag{7}$$

Здесь $\Delta(j)$ – число неисправных вершин, тестирующих j, а $N_b - t$ – число исправных вершин графа.

При выполнении неравенства (7) найдутся такие образ неисправностей и совместный с ним синдром, что каждая исправная вершина является 1-покрытой.

3. Условия тупикового состояния вершины

При анализе условий тупикового состояния неисправной вершины используются лемма 1 и следствие 1, которые дают некоторые общие свойства образа неисправностей, топологически покрывающего граф.

Лемма 1. При $N \ge 2t + 1$ для каждой вершины $j \in F_k$ число тестируемых исправных вершин в $Y(j)$ больше значения $\Delta(j)$.

Доказательство. Согласно конструкции циркулянта, $|Y(k)| = |X(k)| = t$ для каждого $k \in V$. Число неисправных вершин в $Y(j)$ составляет $\delta(j) \le t - (\Delta(j) + 1)$ по условию о кратности неисправностей (равенство имеет место при $N = 2t + 1$ и нечетном N). Следовательно, число исправных вершин в $Y(j)$ будет $t - \delta(j) \ge$ $\ge t - (t - (\Delta(j) + 1)) = \Delta(j) + 1$, что и завершает доказательство.

Следствие 1. Для каждой вершины $j \in F_k$ найдется хотя бы одна 0-покрытая исправная вершина.

Основываясь на отношениях (5) – (6), в самом общем виде признаки тупикового состояния неисправной вершины можно сформулировать в следующем виде.

Свойство 1. Вершина $j \in F_k$ имеет тупиковое состояние, когда для каждого треугольника тестирования (j, k, m) вида A_1 выполняется хотя бы одно из условий а) $a(j, k) = 1$, б) $a(k, m) = a(j, m)$ при том дополнительном условии, что $|Y_1(j)| \le \Delta(j)$.

Выразим свойство 1 в виде совокупности признаков, учитывающих взаимное расположение исправных и неисправных вершин множества $Y(j)$.

Свойство 2. Чтобы для неисправной вершины j достигалось тупиковое состояние, необходимо:

а) все 1-покрытые из j исправные вершины $i \in (V - F_k) \cap Y(j)$ предшествуют 0-покрытым исправным вершинам;

б) все неисправные вершины $j_1 \in Y(j)$, для которых $a(j, j_1) = 0$, предшествуют 0-покрытым исправным вершинам;

в) все неисправные вершины $j_1 \in Y(j)$, которые следуют за 0-покрытыми исправными вершинами, имеют $a(j, j_1) = 1$;

г) для каждой 0-покрытой неисправной вершины $j_1 \in Y(j)$ значение $a(j_1, k)$ совпадает со значением $a(j, k)$, где $k \in Y(j) \bigcap Y(j_1)$, при произвольном (исправное/неисправное) состоянии вершины k.

Пусть для заданных образа неисправностей F_k и совместного с ним синдрома для каждой вершины графа условия (1) – (4) самоопределения не выполняются. Это означает, что для каждой исправной вершины i множество $X_1(i) \ne \varnothing$, а для каждой неисправной вершины j выполняется условие $|X_1(j) \cup Y_1(j)| < t$. Рассмотрим условия тупикового состояния неисправной вершины, вытекающие из невыполнения для нее соотношений (5) – (6) самоопределения с помощью треугольников тестирования вида A_1.

Лемма 2. Чтобы неисправная вершина j не имела определяющих ее треугольников тестирования, необходимо выполнение условия $\exists j_1 \in Y(j) \cap F_k \{a(j, j_1) = 1\}$.

Доказательство. Допустим обратное: $\forall j_1 \in Y(j) \cap F_k \{a(j, j_1) = 0\}$, но j не имеет определяющих треугольников. Из свойства 2, следует, что все 1-покрытые из j исправные вершины i_1 должны предшествовать неисправной вершине j_1, наиболее удаленной от вершины j по связям s_1. Из леммы 1 следует, что существует 0-покрытая из j исправная вершина $i_1 > j_1$, и $i_1 = (j + t) \bmod N$. Из свойства 2-г вытекает, что $\forall j_1 \in Y(j) \cap F_k \{a(j_1, (j + t)) = 0\}$, так как $a(j, (j + t)) = 0$ по условию. Но вершина $(j + t)$ может быть 1-покрыта только из вершин множества $\{j \cup (Y(j) \cap F_k)\}$. Следовательно, при введенных условиях нарушается условие 1-покрытия графа. Значит, при отсутствии 1-покрытых из j неисправных вершин тупиковое состояние вершины j не достигается. Конец доказательства.

Следствие 2. Если неисправная вершина j имеет тупиковое состояние, то $X_0(j) \ne \varnothing$.

Доказательство. Пусть неисправная вершина j 1-покрыта всеми $j_1 \in X(j) \cap F_k$, тогда $X_0(j) = \varnothing$, т.е. $|X_1(j)| = t$, и при $|Y_1(j)| \ge t$ (что имеет место при 1-покрытии из j неисправных вершин) вершина j самоопределима по условию (1).

Нетрудно получить также следующий результат.

Следствие 3. Если для неисправной вершины j при тупиковом состоянии графа выполняется $\Delta(j) = 1$, то вершина j имеет определяющий ее треугольник тестирования.

4. Анализ состояния графа

Для анализа локальной t-диагностируемости графа при заданном образе неисправностей F_k используем свойства треугольников тестирования, построенных на вершинах $\{j, k, (j + t)\}$, где $(j + t) > k > j$ и $j \in F_k$.

Согласно следствию 1, существует $k \in V - F_k$, для которого $a(j, k) = 0$. Если а) $(j + t) \in V - F_k$ и $a(j, (j + t)) = 1$ или б) $(j + t) \in V - F_k$ и $a(j, (j + t)) = 1$, то в соответствии с (5) треугольник тестирования $(j, k, (j + t))$ определяет j. Отсюда необходимое условие, что для $j \in F_k$ нет определяющих треугольников тестирования, имеет следующий вид:

$$\text{если } (j + t) \in V - F_k, \text{ то } a(j, (j + t)) = 0; \tag{8}$$

$$\text{если } (j + t) \in F_k, \text{ то } a(j, (j + t)) = 1. \tag{9}$$

В случае (8) условие 1-покрытия графа требует, чтобы существовала такая неисправная вершина в $Y(j)$, которая 1-покрывает исправную вершину $(j + t)$. Без нарушения условия, что j имеет тупиковое состояние, такой вершиной может быть только неисправная вершина j_1, 1-покрытая из j. Это означает, что достижение тупикового состояния для j уменьшает по крайней мере на две единицы общую сумму $M(F_k) = \sum_{j \in F_k} \Delta(j) - g$. Величина $M(F_k)$ равна наибольшему числу единичных исходов тестирования в синдроме, совместном с заданным образом неисправностей, которая может быть использована на покрытие не-

исправных вершин графа без нарушения условия об отсутствии самоопределимых неисправных вершин. Здесь g – число исправных вершин в графе: $g = N - |F_k|$.

Для случая (9) условие тупикового состояния вершины j означает, что каждая неисправная вершина j_1 из $Y(j)$ должна быть 1-покрыта из вершины j или сама должна 1-покрывать вершину $(j + t)$. Таким образом, в случае (9) достижение тупикового состояния для j уменьшает по крайней мере на $2\delta(j)$ единиц значение $M(F_k)$.

Предложен алгоритм, который использует описанные условия для определения таких совокупностей $f_k \subseteq F_k$ неисправных вершин, тупиковое состояние которых не нарушает условия $\sum_{j\in F_k} \Delta(j) - g \geq \sum_{j\in f_k} q(j)$, что существенно уменьшает трудоемкость анализа локальной диагностируемости при заданном образе неисправностей F_k. Здесь $q(j)$ — число единичных исходов тестирования, используемых для 1-покрытия смежных вершин при тупиковом состоянии вершины j.

5. Локальная t-диагностируемость оптимального циркулянтного графа

Из определения 1-покрытия графа вытекает следующее

Следствие 4. Если $g = \sum_{j\in F_k} \Delta(j)$, то все неисправные вершины $j \in F_k$ имеют определяющие их треугольники тестирования вида A_1.

Более того, можно доказать следующее.

Лемма 3. Если для заданного образа неисправностей $F_k \in F(t)$ выполняется $\sum_{j\in F_k} \Delta(j) \leq g + 4$, то образ неисправностей локально t-диагностируем.

Лемма 3 позволяет установить новую нижнюю границу числа вершин локально t-диагностируемого графа. Обозначим F_{km} образ неисправностей, для которого достигается максимальное значение суммы $M(F_k)$ на множестве $F(N, t)$.

Утверждение 1. Если для заданных N и t при максимальном образе неисправностей F_{km} выполняется $\sum_{j\in F_{km}} \Delta(j) \leq g + 4$, то граф локально t-диагностируем.

Утверждение 1 имеет скорее теоретическое, нежели практическое значение, поскольку для оптимальных графов уже при $t > 6$ имеет место $\sum_{j\in F_{km}} \Delta(j) > g + 4$.

Вместе с тем утверждение 1 позволяет уменьшить число образов неисправностей, требующих исследования на возможность тупикового состояния графа при $\sum_{j\in F_{km}} \Delta(j) > g + 4$. Тем самым уменьшается трудоемкость решения задачи анализа локальной t-диагностируемости заданного графа.

Принципиальную возможность эффективного использования треугольников тестирования показывает следующее.

Лемма 4. В оптимальном циркулянтном графе для каждой неисправной имеется треугольник тестирования вида A_2.

Доказательство. По построению, в циркулянте при $N = 2t + 1$ для всяких k, $k_1 \in V$, таких что $k_1 \in Y(k)$, имеет место $Y(k_1) \cap X(k) \neq \emptyset$. Если $k_1 = k + a$, то $|Y(k_1) \cap X(k)| = a$. Рассмотрим теперь $j \in F_k$ и примем, что для нее нет ни одного определяющего треугольника тестирования вида A_1. Обозначим i исправную вершину в $Y(j)$, $i = j + k$, которая является 0-покрытой исправной вершиной, наиболее удаленной от j по связям с меткой s_1.

Если вершина j имеет тупиковое состояние, то для любых $i_1, i_2 \in [Y(j) \cap (V - F_k)]$, таких, что $i_1 > i_2$, имеет место $\{a(j, i_1) \geq a(j, i_2)\}$. Следовательно, если i – наиболее удаленная от j 0-покрытая исправная вершина, то ни один треугольник тестирования с участием вершины i не будет определяющим для j, если и только если $\{(i + 1), (i + 2), \ldots, (j + t)\} \subset F_k$. С учетом аксиомы о кратности неисправностей отсюда вытекает существование $i_1 \in Y(i) \cap X(j) \cap (V - F_k)$; треугольник тестирования (j, i, i_1) является определяющим для вершины j.

Если в оптимальном циркулянте $N = 2t + 2$, то для каждой вершины $k \in V$ вершина с номером $(k + t + 1)$ не входит в $Y(k) \cup X(k)$. Рассматриваем вершину $j \in F_k$, для которой отсутствуют определяющие треугольники тестирования вида A_1, и вершину $i \in Y(j) \cap (V - F_k)$, являющуюся наиболее удаленной от j по связям с меткой s_1 0-покрытой исправной вершиной. Примем, что $i = j + k$. Из аксиомы кратности неисправностей следует, что общее число неисправных вершин в $Y(i)$ не превышает величины $(t - 1)$. По определению, все вершины, следующие за i, неисправны, так что в $Y(i) - Y(i) \cap Y(j)$ содержится не более $(t - k - 1)$ неисправных вершин. Предположим, что имеет место худший случай, т.е. $|[Y(i) - Y(i) \cap Y(j)] \cap F_k| = t - k - 1$, т.е. $[X(j) - X(j) \cap Y(i)] \cap F_k = \emptyset$.

Имеют место два случая.

1. Если исправная вершина в рассматриваемом множестве $Y(i) - Y(i) \cap Y(j)$ имеет номер $i_1 > j + t + 1$, то существует треугольник тестирования (j, i, i_1), определяющий вершину j.

2. Пусть теперь $i_1 = j + t + 1$. Необходимое условие тупикового состояния вершины j – наличие $Y_1(i) \neq \varnothing$; следовательно, имеется хотя бы одна вершина $j_1 \in F_k \cap Y(i) \cap X(j)$, такая, что $a(j_1, j) = 0$, в таком случае треугольник тестирования $(j, i, j_1,)$ определяет j.

Следствие 5. Оптимальный циркулянтный граф локально t-диагностируем при использовании треугольников тестирования.

Заключение

Рассмотрен системный уровень диагностирования ВС при множественных отказах и исходах тестирования, соответствующих ПМЧ-модели. В качестве диагностической структуры использован циркулянтный граф с вершинной степенью, равной кратности допустимых отказов t. Выявлены и проанализированы условия, когда локальное диагностирование является неполным. Выведены условия достоверной идентификации состояния модуля ВС, производимой только по результатам тестирования из модулей, имеющих с ним непосредственную связь – условия локального t-диагностирования, с помощью сопоставительного анализа фрагментов синдрома, соотнесенных со структурной единицей диагностического циркулянта, названной треугольником тестирования.

Анализ показал, что для перехода вершины в тупиковое состояние вследствие отсутствия определяющих ее треугольников тестирования необходимо одновременное выполнение многочисленных условий, связанных как с взаимным расположением и числом смежных неисправных вершин, так и со значением фрагмента синдрома, соотнесенного с рассматриваемой вершиной. Одно это свидетельствует о том, что доля образов неисправностей, которые могут приводить к тупиковому состоянию графа, при использовании треугольников тестирования значительно сокращается. При этом условия достижения тупикового состояния одними неисправными вершинами могут служить условиями самоопределения для других вершин, в том числе исправных, что в конечном счете приводит к определению состояния всех вершин графа. Многочисленные примеры показали, что использование треугольников тестирования во всех рассмотренных случаях обеспечивает локальную t-диагностируемость графа без использования дополнительного тестирования. Это и доказанная эффективность треугольников тестирования для локальной диагностируемости оптимального циркулянта позволяют выдвинуть гипотезу, что отношения (1) – (6) составляют достаточное условие для достижения локальной t-диагностируемости графа при любых значениях N и t.

Вместе с тем анализ показал, что совместное рассмотрение треугольников тестирования, построенных на множестве $j \cup H(j)$, не дает дополнительной информации об определимости вершин, кроме той, что может быть получена из соотношений (4). Это свидетельствует о том, что для получения новых условий определимости вершины необходимо расширять ее анализируемую окрестность.

ЛИТЕРАТУРА

1. *Димитриев Ю.К.* Локальная самодиагностика вычислительных систем при множественных отказах // Вестник ТГУ. Приложение. 2006. № 17. С. 198 – 202.
2. *Димитриев Ю.К.*, *Задорожный А.Ф.* Условия локального самодиагностирования в вычислительных системах с циркулянтной структурой // Вестник ТГУ. Приложение. 2007. № 23. С. 216 – 220
3. *Preparata F.P.*, *Metze G.*, *Chien R.T.* On the connection assignment problem of diagnosable systems // IEEE Trans. Electr. Comput. 1967. V. EC-16. P. 848 – 854.

2008 Математические основы надежности вычислительных и управляющих систем № 2(2)

DOI 10.17223/20710410/2/22

УДК 519.17: 681.3

ФУНКЦИЯ СТРУКТУРНОЙ ОТКАЗОУСТОЙЧИВОСТИ И *d*-ОГРАНИЧЕННАЯ КОМПОНЕНТА СВЯЗНОСТИ ГРАФА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ[1]

В.А. Мелентьев

Институт физики полупроводников СО РАН, г. Новосибирск

E-mail: melva@isp.nsc.ru

Введено понятие *d*-ограниченной компоненты связности (*d*-компоненты связности) графа вычислительной системы (ВС). Исследованы функции отказоустойчивости кольцевой и полносвязной структур ВС, ограничивающие область определения этой функции для структур, промежуточных в отношении их степени.

Ключевые слова: *структурная отказоустойчивость*, *вычислительная система*, *компонента связности.*

Известно, что повышение надежности системы основано на принципе предотвращения ее неисправностей посредством использования высоконадежных компонентов со сверхвысокой степенью интеграции, резервирования этих компонентов, поддержания облегченных тепловых и вольт-амперных режимов их работы, а также экранированием от воздействия разрушающих внешних импульсных, геомагнитных и радиационных воздействий, совершенствованием методов профилактического обслуживания и сборки аппаратуры и т.п. Понятно, что традиционное использование экспоненциальной модели в оценках надежности составляющих вычислительную систему элементов могло бы быть оправдано лишь для определения максимального числа $l_{\max}$ отказов, ожидаемого от эксплуатации системы в течение всего срока ее службы исключительно в нормативных режимах. Но реальные условия эксплуатации, загруженность системы и качество ее распределения по элементам системы, как показано в работе [1], существенно меняют загруженность системы и ее элементов, а следовательно, надежностные показатели и ожидаемое число исправных из них. Следует учитывать также и то, что наличие минимально необходимого числа исправных элементарных машин (ЭМ) в системе не гарантирует ее работоспособность при утрате требуемых для взаимодействия ЭМ коммуникационных качеств. Отказоустойчивость следует рассматривать как условную вероятность сохранения системой работоспособности на множестве образов ее неисправностей. Как видно из этого определения, свойство отказоустойчивости системы не зависит от показателей надежности составляющих ее элементов и корреспондируется ее архитектурой, нацеленной на сохранение способности выполнения ею в реальном времени и с определенным качеством необходимого минимума функций, достаточного для получения определенного техническими требованиями результата при возникновении любого отказа или их группы в пределах заданной кратности. В работе [2] показано, что отказ от учета функциональной и соответствующей ей структурной составляющих архитектуры системы сводит анализ отказоустойчивости к частному случаю, достоверному лишь для анализа полносвязных систем, где $N' \equiv N - l$, в которых число N' взаимно достижимых ЭМ и число исправных ЭМ тождественны при любой кратности $l < N$ отказов. Понятно, что в теории больших систем, где условие полной связности практически не реализуемо, подобный подход не является достоверным.

В работе [3] автором введены показатели структурной отказоустойчивости и структурной живучести вычислительной системы, определяющие соответственно долю подмножества работоспособных в множестве возможных конфигураций ВС при кратности l отказов и средневзвешенное (на этом множестве) значение определяющего работоспособность показателя качества. Общее число конфигураций системы из N ЭМ при наличии в ней l отказавших ЭМ определяется числом C_N^l сочетаний в группе из N элементов по l. Конфигурация в данном подходе считается работоспособной, если размер соответствующей ей компоненты связности графа ВС не менее предельного числа n элементарных машин, а ее диаметр не превышает заданного значения d. В представленной работе исследована роль структурной составляющей в формировании у ВС свойства отказоустойчивости.

1. Понятия структурной отказоустойчивости ВС и *d*-компоненты связности ее графа

Структурную отказоустойчивость системы из N ЭМ определим отношением числа конфигураций, сохраняющих при заданной кратности l отказов требуемое для успешного функционирования системы минимально необходимое число n элементарных машин, соответствующих заданным критериям связанности, к общему числу C_N^l конфигураций. Минимально необходимое число n элементарных машин обусловлено при

[1] Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект 05-08-01301а).

этом их производительностью, числом и трудоемкостью существенных функциональных подсистем, граничными значениями показателей реактивности системы и плотностью вероятности распределения в ней потока задач [1]. В той же работе [1] показано, что значение кратности $l_{\max}$ допускаемых в ВС отказов определяется не только числом N входящих в ее состав ЭМ, но и ожидаемой загрузкой системы, эффективностью алгоритмов выравнивания нагрузки, а также учитывающим неизбежные отклонения реальных условий эксплуатации элементов системы от нормативных запасом устойчивости к отказам. Таким образом, общее число N ЭМ в системе следует определять, исходя из заданного значения n, из предельного значения $l_{\max}$ кратности допускаемых отказов, причем в общем случае $N \geq n + l_{\max}$, и из структуры сети связи, гарантирующей при значениях N и $l_{\max}$ требуемое число n взаимно достижимых ЭМ. Необходимую для нормального функционирования связанность ЭМ в системе зададим диаметром d, соответствующим допускаемой во взаимодействиях любой пары ЭМ задержке.

Задача анализа структурной отказоустойчивости вычислительной системы представлена здесь в терминах теории графов. В работе [3] автором впервые введено понятие толерантности графа, как способности сохранять заданные свойства при удалении из него некоторого числа вершин. Для исключения громоздких конструкций в последующем тексте уточним некоторые используемые далее определения. Так, компонента связности в теории графов определена максимальным связным подграфом $G'(V', E')$ графа $G(V, E)$. В настоящей работе введем понятие d-ограниченной компоненты связности (d-компоненты связности), выделяющей в графе $G(V, E)$ максимальный связный подграф $G_d'(V_d',E_d')$ с диаметром, не превышающим предельного значения d: $\forall u, v \in V_d'\ d(u, v) \leq d$. Очевидно, что при $d(G') \leq d$ d-компонента связности $G_d'(V_d', E_d')$ совпадает с компонентой $G'(V', E')$ и $G_d'(V_d', E_d') \equiv G'(V', E')$, а при $d(G') > d$ она является ее подграфом $G_d'(V_d', E_d') \in G'(V', E')$.

Наряду с известным из теории графов определением (вершинной) связности $\kappa(G)$ как наименьшего числа вершин в графе $G(V, E)$, удаление которых приводит к несвязному или тривиальному графу, здесь введено понятие d-ограниченной связности $\kappa_d(G)$ графа, определяемой как наименьшее число вершин, удаление которых приводит к появлению в графе d-компоненты связности V_d' с меньшим, чем $|V'|$, размером. Введенные выше определения очевидным образом расширяют известное неравенство Уитни [4] до следующего: $\kappa_d(G) \leq \kappa(G) \leq \lambda(G) \leq \delta(G)$; здесь $\lambda(G)$ и $\delta(G)$ соответственно – значения реберной связности графа и минимальной степени его вершин. Используя введенные определения, обусловим толерантность графа наличием в нем d-компонент связности $G_d'(V_d', E_d') \in G$, размеры $N_d' = |V_d'|$ которых не меньше граничного числа n при заданной кратности l отказов при том, что $N - n \geq l$. Граф G назовем (n, l, d)-толерантным, если для любой из множества конфигураций l-кратных отказов существует d-компонента связности, размер которой не меньше n. Систему, граф G которой (n, l, d)-толерантен, считаем при этом (n, l, d)-структурно отказоустойчивой. Таким образом, задача анализа отказоустойчивости системы сводится к анализу (n, l, d)-толерантности ее графа, равно как и суть задачи синтеза отказоустойчивой структуры ВС состоит в построении соответствующего (n, $l_{\max}$, d)-толерантного графа. Очевидно при этом, что число вершин N ограничено снизу суммой $n + l_{\max}$. Число вершин сверх этой суммы $\Re_V = N - (n + l_{\max})$ будем называть избыточным.

2. Функция структурной отказоустойчивости ВС

Задача оценки структурной отказоустойчивости ВС с критическими значениями числа n вершин и диаметра d в соответствующем ей графе поставлена впервые. Суть ее состоит в выявлении наименьшего числа l^* вершин, удаление которых либо переводит диаметр $d(G')$ компоненты связности в закритическую область – $d(G') > d$, либо при сохранении диаметра в докритической области $d(G') \leq d$ уменьшает число вершин в компоненте более чем на l^*. В обоих случаях $N_d' < N - l^*$, и значение $l^* = \kappa(G)$ является d-ограниченной связностью графа G. Если при этом $l^* > l_{\max}$, т.е. $\forall l \leq l_{\max}\ N_d'(l) = N - l$, то отказоустойчивость системы достигается уже при числе элементарных машин $N = n + l_{\max}$, т.е. при нулевой вершинной избыточности – $\Re_V = 0$. Структурная отказоустойчивость системы в этом случае обусловлена избыточной связностью ее графа. Если же $l^* \leq l_{\max}$, т.е. $\forall l\ |l^* \leq l \leq l_{\max}\ n \leq N'(l) < N - l$, то система также может быть отказоустойчивой, но определяется это свойство уже не только коммуникационной избыточностью, недостаточной для $\Re_V = 0$, – система должна быть дополнена также некоторым избыточным числом ЭМ, т.е. должно быть $N > n + l_{\max}$ и $\Re_V > 0$.

Показателен в этом отношении анализ двух полярных с позиций избыточности регулярных структур: кольцевой, где число ребер $|E|$ минимально и не превышает $N - 1$, и полносвязной структуры (K_n-графа), содержащей максимально возможное число ребер – $|E| = N(N - 1)/2$. Реберную избыточность $\Re_E$ кольцевой структуры с $N = n + l_{\max}$ примем равной нулю, а число ребер сверх $n + l_{\max} - 1$ назовем избыточным.

Рассмотрим в качестве примера систему с кольцевой структурой $G(V,E)$, в которой заданы минимально необходимое для сохранения работоспособности число n вершин и их изначальное число $N \geq n$. Считаем, что одновременно в системе может быть задействовано максимально возможное число $N_d' = |V'|$, $n \leq N_d' \leq N$, элементарных машин в d-компоненте связности $G_d'(V_d',E_{V_d'}) \subseteq G'(V',E_{V'})$ с диаметром $d(G_d')$, не превышающим величины d из замкнутого промежутка $[n - 1, N - 1]$. В соответствии с [3] значение структурной отка-

зоустойчивости $\theta(n, l, d)$ такой системы при кратности l отказов определяем отношением

$$\theta(n, l, d) = (C_N^l)'/C_N^l .$$

Так как в кольцевом графе значение d однозначно определено числом n ($d = n - 1$), то этот параметр далее можно опустить. Здесь $(C_N^l)' = |\{G_i(l)\} \wedge \mathrm{H}|$, $\{G_i(l)\}$ – множество подграфов графа G, соответствующее множеству сочетаний по l удаленных вершин графа из всех N вершин, $i = \overline{1, C_N^l}$, H – предикат адекватности, устанавливающий соответствие (H = 1) или несоответствие (H = 0) исследуемого i-го подграфа $G_i(l)$ заданным выше условиям работоспособности ВС:

$$\exists G_i' \in G_i(l) \,|\, d(G_i') \le d, |V_i'| \ge n \Rightarrow G_i(l) \wedge \mathrm{H} = 1.$$

Если $(C_N^l)' = C_N^l$, то $\theta(n, l) = 1$ и, следовательно, кольцевая система (n, l)-структурно отказоустойчива. При $(C_N^l)' < C_N^l$ ее структурная отказоустойчивость $\theta(n, l) < 1$ и существует отличная от нуля вероятность утраты существенных структурных качеств, необходимых для ее работоспособности.

Очевидно, что в кольцевой ВС с $N > n$ при кратности отказов $l = 1$ всегда справедливо $(C_N^1)' = C_N^1 = N$ и ее структурная отказоустойчивость $\theta(n, 1) = 1$. Условием безусловной структурной отказоустойчивости к кратности $l = 2$ отказов, при котором $(C_N^2)' = C_N^2$, является $\lceil (N-2)/2 \rceil \ge n$, или $N \ge 2n + 1$. Условием же безусловной структурной отказоустойчивости кольцевой ВС к произвольной кратности отказов $l \ge 1$ является $N \ge nl + 1$. Очевидно, что это условие является также определяющим при синтезе отказоустойчивых кольцевых ВС и сетей связи. Итак, в кольцевой структуре $N \ge nl + 1 \Rightarrow \theta(n, 1) = 1$.

В табл. 1 даны результаты расчета структурной отказоустойчивости кольцевой ВС с достаточным для ее нормального функционирования значением $n = 10$, $l = \overline{2,5}$.

Перечислить работоспособные конфигурации $(C_N^l)'$ в кольце достаточно просто: вычислив $\lceil (N-l)/l \rceil$, получим значение минимально возможного размера компоненты связности кольцевого графа при удалении из него l вершин. Покажем это на примере $N = 15$, $l = 3$, $n = 10$: $\lceil (15-3)/3 \rceil = 4$, $C_{15}^3 = 455$. В табл. 1 приведены размеры подграфов, получаемых при удалении 3-х вершин. Здесь N_1 – размер компоненты связности, N_2 и N_3 – размеры других изолированных подграфов. Затененная в таблице область соответствует комбинациям, размер компоненты связности в которых превышает $n = 10$. Так как общее число таких комбинаций $(C_{15}^3)' = 90$, то $\theta(10, 3) = (C_{15}^3)'/C_{15}^3 = 90/455 = 0{,}1978$.

Таблица 1

№ п/п	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Число комбинаций	15	30	30	15	30	30	30	30	15	30	30	30	15	30	30	15	15	30	5
N_1	12	11	10	10	9	9	8	8	8	7	7	7	6	6	6	6	5	5	4
N_2	0	0	0	1	0	1	0	1	2	0	1	2	0	1	2	3	2	3	4
N_3	0	1	2	1	3	2	4	3	2	5	4	3	6	5	4	3	5	4	4

В табл. 2 сведены данные расчетов функций структурной живучести рассматриваемого кольцевого графа от l (при $n = 10$), представленные на рис. 1.

Таблица 2

l	***N***	***n+l***	**15**	**20**	**21**	**25**	**30**	**31**	**35**	**40**	**41**	**45**
	C_N^2	66	105	190	210							
2	$C_2'(N)$	12	60	180	210							
	$\theta(10,2)$	0,182	0,571	0,9474	1,0							
	C_N^3	286	455	1140	1330	2300	4060	4495				
3	$C_3'(N)$	13	90	720	945	2125	4050	4495				
	$\theta(10,3)$	0,045	0,198	0,632	0,711	0,924	0,998	1,0				
	C_N^4	1001	1365	4845	5985	12650	27405	31465	52360	91390	101270	
4	$C_4'(N)$	14	60	1680	2520	9025	25305	29791	51800	91350	101270	
	$\theta(10,4)$	0,014	0,044	0,347	0,421	0,713	0,923	0,947	0,989	0,9996	1,0	
	C_N^5		3003	15504	20349	53130	142506	169911	324632	658008	749398	1221759
5	$C_5'(N)$		15	1420		15375	91141		291656		744355	1221759
	$\theta(10,5)$		0,005	0,092		0,289	0,64		0,898		0,993	1,0

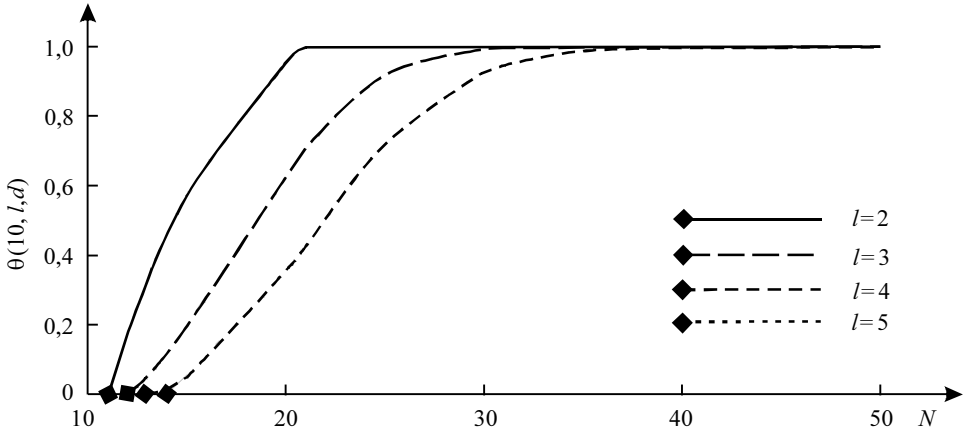

Рис. 1. Структурная отказоустойчивость кольцевой ВС с $n = 10$

Система, все ЭМ которой связаны по полному графу, отказоустойчива уже при $N = n + l_{\max}$, но число ребер в графе такой ВС увеличивается в сравнении с кольцевой структурой до $N(N-1)$. Скачкообразное (при $N = n + l$) изменение значения функции структурной отказоустойчивости полносвязной ВС от $\theta(n, l) = 0$ до $\theta(n, l) = 1$ видно из графика этой функции при $n = 10$, $l = \overline{2,5}$ (рис. 2). Здесь же выделена область, представляющая собой разность функций структурной отказоустойчивости кольцевой и полносвязной ВС для $n = 10$, $l = 5$. Совершенно очевидно, что значения структурной живучести систем с соответственно равными значениями n и l, но с разными значениями вершинной связности $2 \le \kappa_d \le N-1$ лежат в построенной таким образом промежуточной области. Из рис. 2 видно, что, например, для $n = 10$, $l = 5$ при $N = 25$ структурная отказоустойчивость системы в зависимости от выбранного для нее графа заключена в пределах от $\theta(10, 5) = 0{,}289$ при средней вершинной связности, близкой к двум ($\kappa \to 2$), до $\theta(10, 5) = 1{,}0$ при $\kappa \to N-1$.

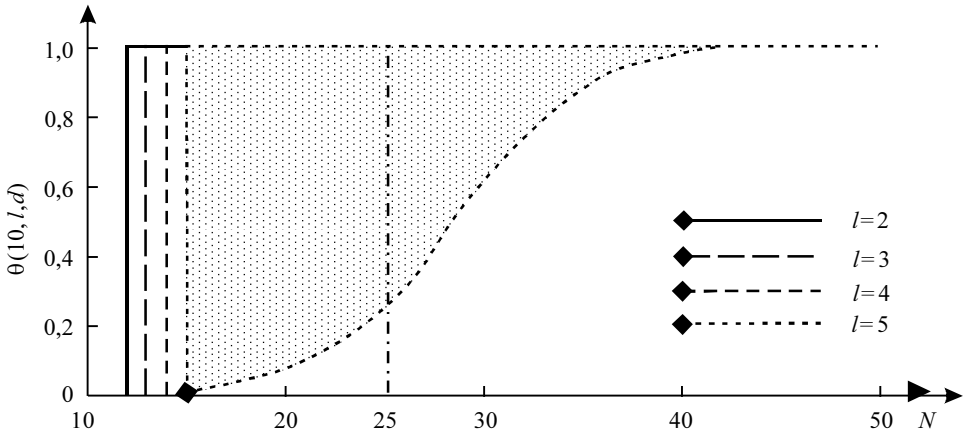

Рис. 2. Структурная отказоустойчивость полносвязной ВС и область определения

Диаметры кольцевого и полного графов существенно рознятся между собой: если в первом случае диаметр не менее чем в $l_{\max}$ раз превышает нижний предел числа ЭМ в работоспособной ВС, то диаметр компонент связности полного графа при любых кратностях $l \ge N - n$ и конфигурациях отказов всегда равен единице. В силу ограниченности допускаемого диаметра в первом случае может быть использована лишь небольшая часть исправных ЭМ, что указывает на низкую эффективность использования суммарной мощности ВС. Число ребер $|E|$ в рассматриваемых кольцевом и полном графах составляет соответственно: $|E_c| = nl_{\max}$ и $|E_n| = (n + l_{\max})(n + l_{\max} - 1)$. Таким образом, значения вершинной и реберной избыточностей для рассмотренных здесь случаев составляют: в системе с кольцевой структурой – $\Re_V = \Re_E = (n-1)(l_{\max} - 1)$, а в полном графе – $\Re_V = 0$ и $\Re_E = (n + l_{\max} - 1)^2$.

Обозначив удельные стоимости вершин C_V и ребер C_E, переведем их в условные единицы: $c_V = C_V/(C_V + C_E)$, $c_E = C_E/(C_V + C_E)$; таким образом, $c_V + c_E = 1$ и в общем случае условная стоимость $c(G)$ реализации структурной отказоустойчивости ВС составляет: $c(G) = c_V\Re_V + c_E\Re_E$. Сопоставление избыточных стоимостей различных вариантов дает необходимые формальные основания для выбора той или иной структуры. В рассмотренных выше случаях, повышающих до предела роль одной из составляющих (вершинной или реберной) за счет другой, условная стоимость отказоустойчивости составляет: для кольцевой ВС –

$C = c_V n(l_{\max} - 1) + c_E n(l_{\max} - 1) = n(l_{\max} - 1)$ и для объединения ЭМ по полному графу – $C = c_E(n + l_{\max} - 1)^2 = (1 - c_V)(n + l_{\max} - 1)^2$. Очевидно, что оптимальная в отношении стоимости отказоустойчивая структура также может быть найдена среди промежуточных вариантов.

Заключение

Структурная отказоустойчивость системы определена сохранением требуемого для успешного функционирования системы минимально необходимого числа взаимосвязанных (соответствующих заданным критериям связанности) ЭМ при любой конфигурации отказов с максимально допускаемой кратностью. Показатель структурной отказоустойчивости вычислительной системы определяет таким образом долю подмножества работоспособных в множестве возможных конфигураций ВС при кратности l отказов. При этом минимально допустимое при отказах число ЭМ определяется исходя из их производительности, из набора существенных функциональных подсистем и их трудоемкости, из граничных значений показателей реактивности системы в решении этих задач, из плотности вероятности распределения потока задач и т.д.

Постановка задачи анализа структурной отказоустойчивости вычислительной системы впервые рассматривает в качестве критических параметров работоспособности минимально допустимый размер n компоненты связности графа ВС и ее предельный диаметр d. В связи с постановкой введено понятие d-ограниченной компоненты связности (d-компоненты связности), выделяющей в графе ВС максимальный связный подграф с диаметром, не превышающим предельного значения d, и понятие d-ограниченной связности графа, определяемой как наименьшее число l вершин, удаление которых приводит к появлению в графе d-компоненты связности с меньшим, чем у компоненты связности, размером. Синтез отказоустойчивой системы при этом заключается в определении кратности l отказов, при которой система должна сохранять работоспособность независимо от конфигурации отказавших ЭМ, и в выборе обеспечивающей это условие структурной конъюнктуры: изначального числа исправных элементарных машин и их связности. Максимальное значение кратности $l_{\max}$ отказов определяется при этом коррелированными ожидаемыми условиями эксплуатации надежностными показателями используемых ЭМ, их минимально допустимым числом и планируемыми значениями вершинной и реберной избыточности исходного графа. Говоря об условиях эксплуатации, мы включаем сюда как внешние (температурные, радиационные и т.п.), так и внутренние факторы (загруженность, качества алгоритмов перераспределения нагрузки и т.п.).

В работе дано определение соответствующей указанной постановке (n, l, d)-структурно отказоустойчивой системы, проведено исследование полярных в отношении связности кольцевой и полносвязной структур. Показано, что при заданных значениях n, l и d область размещения семейства функций структурной отказоустойчивости систем с варьируемыми соотношениями между $\Re_V$ и $\Re_E$ ограничена снизу кольцевой ($\Re_E = 0$) и сверху полносвязной ($\Re_V = 0$) структурами. Введенный в работе показатель стоимости, выраженный в условных единицах относительной стоимости вершинной и реберной избыточности структуры ВС, может быть использован в качестве критерия структурной оптимизации близких по значению структурной (n, l, d)-отказоустойчивости систем.

ЛИТЕРАТУРА

1. *Мелентьев В.А.* Корреляция надежности элементов вычислительной системы реальными условиями ее эксплуатации // Вестник ТГУ. Приложение. 2007. № 23. С. 247 – 252.
2. *Мелентьев В.А.* Обобщенная модель отказоустойчивой системы // Вестник ТГУ. Приложение. 2007. № 23. С. 242 – 246.
3. *Мелентьев В.А.* Толерантность графов и структурная отказоустойчивость вычислительных систем // Вестник ТГУ. Приложение. 2004. № 9(I). С. 144 – 150.
4. *Харари Ф.* Теория графов. М.: Мир, 1973.

2008 Математические основы надежности вычислительных и управляющих систем № 2(2)

DOI 10.17223/20710410/2/23

УДК 519.17: 681.3

ПОИСК ВЕРШИННЫХ (s, t)-СЕЧЕНИЙ ГРАФА ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ С ОГРАНИЧЕНИЕМ ПО ДИАМЕТРУ КОМПОНЕНТ СВЯЗНОСТИ[1]

В.А. Мелентьев

Институт физики полупроводников СО РАН, г. Новосибирск

E-mail: melva@isp.nsc.ru

В рамках задачи поиска d-ограниченной связности графа вычислительной системы (ВС) предложен подход к решению проблемы поиска множества минимальных d-ограниченных (s, t)-сечений, базирующийся на скобочной форме представления проекций и образов графа.

Ключевые слова: *структурная отказоустойчивость вычислительной системы*, *диаметр компоненты связности.*

Отказоустойчивость системы базируется на ее структурной конъюнктуре, заключающейся в наличии достаточного для сохранения работоспособности числа вершин в графе ВС и в обеспечении заданного критерия их связанности. В качестве последнего здесь определено значение максимально допускаемого в системе эксцентриситета вершин, или диаметра компоненты связности графа ВС при ее деградации вследствие происходящих в системе отказов. Задача оценки структурной отказоустойчивости ВС с критическими значениями числа n вершин и диаметра d впервые поставлена в [1]. Как указано в [2], суть этой задачи состоит в выявлении наименьшего числа l^* вершин, удаление которых переводит диаметр компоненты связности в закритическую область $d(G') > d$, или при сохранении диаметра в докритической области $d(G') \le d$ уменьшает число вершин в компоненте более чем на l^*. В обоих случаях $N' < N - l^*$, и значение $l^* = \kappa_d(G)$ является d-ограниченной связностью графа G. Если при этом $l^* > l_{\max}$, т.е. $\forall l \le l_{\max}$, $N' = N - l$, то отказоустойчивость системы достигается уже при $N = n + l_{\max}$ элементарных машин, т.е. при нулевой вершинной избыточности $\Re_V = N - (n + l_{\max})$. Здесь $l_{\max}$ – максимально допускаемое в системе значение кратности отказов, при которой существенные для ее работоспособности критерии качества находятся в заданных пределах. Структурная отказоустойчивость системы в этом случае достигается только за счет избыточной связности ее графа. Если же $l^* \le l_{\max}$, т.е. $\forall l \ge l^* \le l_{\max}$, $n \le N'(l) < N - l$, то система также отказоустойчива, но достигается это свойство не только за счет коммуникационной избыточности, недостаточной для $\Re_V = 0$: система должна быть дополнена некоторым избыточным числом ЭМ, т.е. должно быть $N > n + l_{\max}$ и $\Re_V > 0$.

Итак, одной из первых задач, возникающих при анализе структурной отказоустойчивости системы, является задача поиска d-ограниченной связности ее графа – $l^* = \kappa_d(G)$. Решение этой задачи базируется на поиске множества минимальных (s, t)-сечений графа $G(V, E)$, представляющих собой $\forall s \in V$, $\forall t \in V \setminus \mathcal{N}[s]$ минимальное множество вершин $V^* \in V_{-s,t}$, удаление которых приводит к появлению d-компоненты связности $G'_d \in G(V \setminus V^*, E \setminus E^*_{V^*})$ такой, что $s \notin G'_d \Rightarrow t \in G'_d$ и $s \in G'_d \Rightarrow t \notin G'_d$, причем $d(s, t) > d$. Здесь $\mathcal{N}[s] = \mathcal{N}(s) \cup s$ – замкнутая окрестность вершины s, а $\mathcal{N}(s)$ – ее окрестность. Поиск минимального d-ограниченного вершинного (s, t)-сечения графа $G(V, E)$ основан на использовании аппарата скобочных образов графа, предложенного автором в [3], и операций над образами, описанных в [4]. В работе предложен и продемонстрирован на примерах подход к реализации такого поиска.

Поиск вершинного (s, t)-сечения графа ВС

Проблема поиска (s, t)-сечений достаточно широко представлена в теории графов. В основе алгоритмизации поиска лежит, как правило, теорема Форда – Фалкерсона [5], в соответствии с которой максимальный поток между вершинами t и s равен величине минимального сечения между этими вершинами. Вероятно, для решения задачи, в постановку которой нами введено ограничение на длину простого пути между s и t, можно приспособить одну из известных модификаций алгоритма поиска максимального потока, но в данной работе мы используем введенный автором в [6] и достаточно хорошо показавший себя в [7, 8] аппарат скобочных образов.

Итак, структура изначально работоспособной ВС задана неориентированным невзвешенным связным графом $G(V, E)$, в котором множеству V вершин поставлено в однозначное соответствие множество ЭМ, а множеству E ребер также однозначно поставлены в соответствие линии связи между ЭМ. Граф $G(V, E)$

[1] Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект 05-08-01301а).

представим его проекцией $P_k(v^0 \in V)$; вершина v^0 здесь является ракурсной, а индекс k определяет число уровней проекции.

Напомним, что в соответствии с [6] образ графа $G(V, E)$ определен вектором $\mathbf{P}(G) = \{P(v_i) \,|\, v_i \in V\}$, представляющим собой множество его проекций $P(v_i)$ в множестве ракурсов $\{v_i \in V\}$. Проекцию $P(v_i)$ графа $G(V, E)$ с ракурсной вершиной $v_i \in V$ называем v_i-й проекцией, либо v_i-м ракурсом этого графа. Полная проекция $P(v_i)$ графа $G(V, E)$ содержит всю необходимую для его построения информацию. Описание проекции представлено в виде конечного множества уровней. Ракурсная вершина (здесь $v^0 \equiv s$) помещена на нулевом уровне, смежные ей вершины заключены внутри скобок 1-го уровня и т.д. Таким образом, 1-й уровень проекции образа графа включает в себя окрестность начальной вершины – $\mathcal{N}(v^0)$. Правило включения множества потомков произвольной j-й вершины i-го уровня в (i+1)-й уровень проекции графа определено выражением … $v_j^i(\{\mathcal{N}(v_j^i) \setminus C(v_j^i)\})$, а совокупность S^{i+1} вершин любого $(i+1)$-го уровня проекции определяется выражением $S^{i+1} = (\{\mathcal{N}(v_1^i) \setminus C(v_1^i)\}, \{\mathcal{N}(v_2^i) \setminus C(v_2^i)\}, \ldots, \{\mathcal{N}(v_{|S^i|}^i) \setminus C(v_{|S^i|}^i)\})$ [6]. Здесь v_j^i – произвольная j-я вершина из совокупности вершин i-го уровня проекции; $j = \overline{1,|S^i|}$, а $|S^i|$ – число элементов совокупности S^i; $\mathcal{N}(v_j^i)$ – окрестность вершины v_j^i; $C(v_j^i)$– множество вершин в составе простой цепи между начальной ($v^0 \equiv s$) и j-й (v_j^i) вершиной i-го уровня, включающее v^0 и v_j^i.

Далее будут использованы следующие обозначения: $M_d(s, t) = \{M_d^i(s, t)\}$ – множество d-ограниченных (s, t)-маршрутов (простых путей между s и t), при этом по умолчанию концевые вершины в множествах вершин, составляющих эти маршруты, опущены, а индекс i идентифицирует маршрут в множестве $M_d(s, t)$: $i = \overline{1,|M_d(s,t)|}$; $V_d(s, t) = \{v_j\} = \bigcup_{i=1}^{|M_d(s,t)|} V_d^i(s,t)$ – множество вершин из $M_d(s, t)$, $V_d^i(s, t)$ – множество вершин из $M_d^i(s, t)$; $S_d(s, t) = \{S_d^i(s, t)\}$ – множество минимальных и равномощных d-ограниченных (s, t)-сечений графа, $S_d^i(s, t)$ – множество вершин, составляющих i-е сечение из $S_d(s, t)$; m_j – кратность вершины $v_j \in V_d(s, t)$, равная числу содержащих эту вершину маршрутов из $M_d(s, t)$.

Отметим некоторые используемые при поиске положения, справедливые при $d(s, t) \le d$ и не требующие доказательства из-за их очевидности.

1. Множество вершинных (s, t)-сечений пусто при $d(s, t) = 1$ (вершины s и t смежны): $d(s, t) = 1 \Rightarrow S_d(s, t) = \varnothing$. Учитывая это, из множества вершин, которые составляют пары с вершиной s и для которых допустимо искать сечение, можно изначально исключить $\mathcal{N}[s]$-замкнутую окрестность вершины s.

2. $\forall S^i(s, t) \in S(s, t)\ \{\mathcal{N}(s) \cap \mathcal{N}(t)\} \in \{S^i(s, t)\}$. Здесь индекс ограничения по диаметру d опущен, так как утверждение справедливо вне зависимости от ограничивающего связанность диаметра. Заметим, что $\mathcal{N}(s) \cap \mathcal{N}(t) \ne \varnothing$ только при $d(s, t) = 2$. В этом случае множество вершин, составляющих любое из множеств всех (s, t)-сечений графа, обязательно включает в себя вершины пересечения окрестностей $\mathcal{N}(s)$ и $\mathcal{N}(t)$. Это позволяет производить поиск d-ограниченных (s, t)-сечений графа, изначально включив вершины пересечения в составы искомых сечений и исключив из $M_d(s, t)$ все маршруты, содержащие эти вершины, соответствующим образом скорректировав при этом множество $V_d(s, t)$ и кратности m_j составляющих его вершин $v_j \in V_d(s, t)$.

3. Если сумма кратностей вершин подмножества из множества $V_d(s, t)$ не превышает числа маршрутов $|M_d(s, t)|$, то таких вершин недостаточно для d-ограниченного (s, t)-рассечения графа. Это утверждение дает основания для первоначального в процессе поиска выбора удаляемой из графа вершины: 1) связность $\kappa_d(G)$ не может быть меньше наименьшего числа вершин из $V_d(s, t)$, сумма кратностей которых превосходит число маршрутов в множестве $M_d(s, t)$; 2) кратность очередной выбираемой в процессе поиска из $V_d'(s,t)$ вершины не может быть менее округляемого вверх отношения $\lceil M_d'(s, t) / \kappa_d(G) \rceil$.

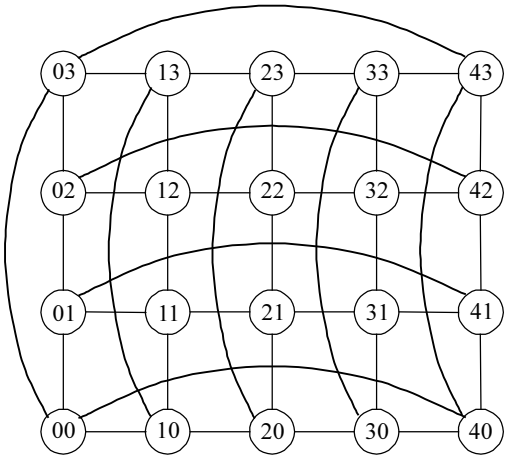

Рис. 1. ВС с тороидальной структурой

На рис. 1 представлена тороидальная структура ВС, на примере которой будет продемонстрирован предлагаемый здесь подход к поиску минимальных (s, t)-сечений. Эта структура достаточно распространена в практике создания ВС из-за удобства в адресной организации и простоты алгоритмов маршрутизации. Помимо этого, в подобных этой структурах можно ограничиться исследованием коммуникационных свойств лишь $(N-1)$-й пары вершин из C_N^2 возможных.

В [6] доказано, что число k уровней проекции, необходимое для полного описания графа, равно эксцентриситету $e(v^0)$ ракурсной вершины v^0, если множество вершин, равноудаленных от нее на величину максимального эксцентриситета, не является связанным, и $k = e(v^0) + 1$, если хотя бы одна пара вершин из этого множества смежна. Заметим, что значение эксцентриситета $e(v^0)$ не требует вычислений, предваряющих процесс построения проекции, а получается в нем автоматически как номер уровня, которым множество вершин, включенных в предшествующие уровни, дополняется до V. В нашем примере для полноты описания было бы достаточно 4-х уровней ($d(G) = 4$), но так как принятое нами выше ограничение на диаметр компоненты связности $d = 5$, то проекцию графа из вершины 00 представим 5-ю уровнями:

$$P_5(00) =$$

00(01(02(03(13(10, 12, 23), 43(33, 40, 42)), 12(11(10, 21), 13(03, 10, 23), 22(21, 23, 32)), 42(32(22, 31, 33), 41(31, 40), 43(03, 33, 40))), 11(10(13(03, 12, 23), 20(21, 23, 30)), 12(02(03, 42), 13(03, 10, 23), 22(21, 23, 32)), 21(20(10, 23, 30), 22(12, 23, 32), 31(30, 32, 41))),
41(31(21(11, 20, 22), 30(20, 33, 40), 32(22, 33, 42)), 40(30(20, 31, 33), 43(03, 33, 42)), 42(02(03, 12), 32(22, 31, 33), 43(03, 33, 40)))),

03(02(01(11(10, 12, 21), 41(31, 40, 42)), 12(11(01, 10, 21), 13(10, 23), 22(21, 23, 32)), 42(32(22, 31, 33), 41(01, 31, 40), 43(33, 40))), 13(10(11(01, 12, 21), 20(21, 23, 30)), 12(02(01, 42), 11(01, 10, 21), 22(21, 23, 32)), 23(20(10, 21, 30), 22(12, 21, 32), 33(30, 32, 43))),
43(33(23(13, 20, 22), 30(20, 31, 40), 32(22, 31, 42)), 40(30(20, 31, 33), 41(01, 31, 42)), 42(02(01, 12), 32(22, 31, 33), 41(01, 31, 40)))),

10(11(01(02(03, 12, 42), 41(31, 40, 42)), 12(02(01, 03, 42), 13(03, 23), 22(21, 23, 32)), 21(20(23, 30), 22(12, 23, 32), 31(30, 32, 41))), 13(03(02(01, 12, 42), 43(33, 40, 42)), 12(02(01, 03, 42), 11(01, 21), 22(21, 23, 32)), 23(20(21, 30), 22(12, 21, 32), 33(30, 32, 43))),
20(21(11(01, 12), 22(12, 23, 32), 31(30, 32, 41)), 23(13(03, 12), 22(12, 21, 32), 33(30, 32, 43)), 30(31(21, 32, 41), 33(23, 32, 43), 40(41, 43)))),

40(30(20(10(11, 13), 21(11, 22, 31), 23(13, 22, 33)), 31(21(11, 20, 22), 32(22, 33, 42), 41(01, 42)), 33(23(13, 20, 22), 32(22, 31, 42), 43(03, 42))), 41(01(02(03, 12, 42), 11(10, 12, 21)), 31(21(11, 20, 22), 30(20, 33), 32(22, 33, 42)), 42(02(01, 03, 12), 32(22, 31, 33), 43(03, 33))),
43(03(02(01, 12, 42), 13(10, 12, 23)), 33(23(13, 20, 22), 30(20, 31), 32(22, 31, 42)), 42(02(01, 03, 12), 32(22, 31, 33), 41(01, 31)))))).

Отсутствие здесь возможностей цветовой разметки уровней проекции затрудняет ее структурное восприятие, поэтому 4 части, каждая из которых содержит в себе одну из вершин окрестности начальной вершины, выделены интервально, а разбиение этих частей на строки выделяет вершины 2-го уровня проекции с их потомками.

Множество вершин, составляющих пары с вершиной 00, для которых допустимо искать сечение, не может включать в себя замкнутую окрестность вершины 00 и представляет собой $V \setminus \mathcal{N}[00] = \{20, 30, 11, 21, 31, 41, 02, 12, 22, 32, 42, 13, 23, 33, 43\}$. Продемонстрируем процесс поиска минимального сечения для пары, составленной из вершины 00 и одной из вершин этого множества, к примеру, (00, 32)-сечения. Из того, что множество возможных сечений в графе включает в себя N окрестностей каждой из вершин графа и в соответствии с расширенным в [2] неравенством Уитни, $\kappa_d(G) \le \kappa(G) \le 4$.

Трансформируя проекцию $P_5(00)$ графа удалением из нее всех концевых вершин, кроме 32-й, получим проекцию $P_5(00, 32)$, содержащую множество простых путей к этой вершине из вершины 00, длина которых не превышает 5; число таких путей в нашем графе равно 31:

$00(_1 01(_2 02(_3 12(_4 22(_5 32_5)_4), 42(_4 32_4)_3), 11(_3 12(_4 22(_5 32_5)_4), 21(_4 22(_5 32_5), 31(_5 32_5)_4)_3), 41(_3 31(_4 32_4), 42(_4 32_4)_3)_2), 03(_2 02(_3 42(_4 32_4)_3), 13(_3 12(_4 22(_5 32_5)_4), 23(_4 22(_5 32_5), 33(_5 32_5)_4)_3), 43(_3 33(_4 32_4), 42(_4 32_4)_3)_2), 10(_2 11(_3 12(_4 22(_5 32_5)_4), 21(_4 22(_5 32_5), 31(_5 32_5)_4)_3), 13(_3 12(_4 22(_5 32_5)_4), 23(_4 22(_5 32_5), 33(_5 32_5)_4)_3), 20(_3 21(_4 22(_5 32_5), 31(_5 32_5)_4), 23(_4 22(_5 32_5), 33(_5 32_5)_4), 30(_4 31(_5 32_5), 33(_5 32_5)_4)_3)_2),$
$40(_2 30(_3 31(_4 32_4), 33(_4 32_4)_3), 41(_3 31(_4 32_4), 42(_4 32_4)_3), 43(_3 33(_4 32_4), 42(_4 32_4)_3)_2)_1).$

Здесь для удобства визуального восприятия концевая вершина выделена затенением, а открывающиеся/закрывающиеся скобки снабжены индексом соответствия их содержимого номеру уровня (длине маршрута до любой вершины уровня). Перечислим полученные пути в явном виде, оставив в множестве $M_5(00, 32)$ – индекс здесь соответствует предельной длине маршрутов, лишь транзитные вершины:

{01-02-12-22, 01-02-42, 01-11-12-22, 01-11-21-22, 01-11-21-31, 01-41-31, 01-41-42,
03-02-42, 03-13-12-22, 03-13-23-22, 03-13-23-33, 03-43-33, 03-43-42,
10-11-12-22, 10-11-21-22, 10-11-21-31, 10-13-12-22, 10-13-23-22, 10-13-23-33, 10-20-21-22, 10-20-21-31, 10-20-23-22, 10-20-23-33, 10-20-30-31, 10-20-30-33,
40-30-31, 40-30-33, 40-41-31, 40-41-42, 40-43-33, 40-43-42}.

Сформируем множество $V_5(00, 32) = \{v_j(m_j)\}$ транзитных вершин $v_j \in M_5(00, 32)$, входящих в состав $M_5(00, 32)$, и выявим кратности m_j их использования – $v_j(m_j)$:

$V_5(00, 32)$ = {01(7), 02(3), 03(6), 10(12), 11(6), 12(5), 13(6), 20(6),21(6), 22(11), 23(6), 30(4), 31(7), 33(7), 40(6), 41(4), 42(6), 43(4)}.

Как указано выше, для получения сечения из трех вершин необходимо, как минимум, чтобы сумма кратностей удаляемых вершин была не менее числа маршрутов. Поскольку таковых в нашем случае нет (12 + 11 + 7 < 31) и минимальный набор вершин, сумма кратностей которых превышает число маршрутов, состоит из 4 вершин, то мощность минимального сечения не может быть менее 4. Удалив используемую в 12 маршрутах вершину 10 и соответствующие ей маршруты, модифицируем кратности использования вершин в оставшихся 19 путях:

{01-02-12-22, 01-02-42, 01-11-12-22, 01-11-21-22, 01-11-21-31, 01-41-31, 01-41-42,
03-02-42, 03-13-12-22, 03-13-23-22, 03-13-23-33, 03-43-33, 03-43-42,
40-30-31, 40-30-33, 40-41-31, 40-41-42, 40-43-33, 40-43-42}

01(7), 02(3), 03(6), 11(3), 12(3), 13(3), 20(0),21(2), 22(5), 23(2), 30(2), 31(4), 33(4), 40(6), 41(4), 42(6), 43(4).

Для рассечения оставшихся 19 маршрутов тремя вершинами кратность использования следующей удаляемой из этих маршрутов вершины должна быть не менее 7. Такой вершиной является вершина 01, после удаления которой число оставшихся маршрутов уменьшится до 12. Модифицируем множество оставшихся маршрутов и кратности используемых в них вершин; вершины с кратностью менее $\lfloor 19/3 \rfloor = 6$ исключаем из рассмотрения:

{03-02-42, 03-13-12-22, 03-13-23-22, 03-13-23-33, 03-43-33, 03-43-42,
40-30-31, 40-30-33, 40-41-31, 40-41-42, 40-43-33, 40-43-42}

03(6), 40(6), 42(4).

Удаление вершин 03 и 40 завершает процедуру поиска для выбора вершины 10 в качестве 1-й удаляемой. В результате поиска найдено сечение с мощностью, равной 4: {10, 12, 03, 40}. Вернувшись к выбору 1-й удаляемой вершины, начнем теперь с удаления вершины 22 с кратностью, равной 11. Описание всех оставшихся при этом маршрутов из вершины 00 в вершину 32 получим аналогично вышеописанному:

{01-02-42, 01-11-21-31, 01-41-31, 01-41-42,
03-02-42, 03-13-23-33, 03-43-33, 03-43-42,
10-11-21-31, 10-13-23-33, 10-20-21-31, 10-20-23-33, 10-20-30-31, 10-20-30-33,
40-30-31, 40-30-33, 40-41-31, 40-41-42, 40-43-33, 40-43-42}

01(4), 02(2), 03(4), 10(6), 11(2), 12(0), 13(2), 20(4),21(3), 22(0), 23(3), 30(4), 31(7), 33(7), 40(6), 41(4), 42(6), 43(4).

Очевидно, что для получения сечения с мощностью, равной 4, кратности следующих двух удаляемых вершин из оставшихся 20 маршрутов должны быть не меньшими 7, а кратность последней из удаляемых при этом не может быть менее 6. Первому из этих условий удовлетворяют только вершины 31(7), 33(7), – удалим их:

{01-02-42, 01-41-42,
03-02-42, 03-43-42,
40-41-42, 40-43-42}

10(0), 40(2), 42(6).

Осталась единственная вершина 42 с кратностью, не меньшей числа оставшихся маршрутов. Удалив ее, получим искомое сечение: {22, 31, 33, 42}. Как видим, множество минимальных (00, 32)-сечений включает 2 сечения мощности 4, совпадающие с окрестностями разделяемых ими вершин. Чтобы показать, что совпа-

дение множества (s, t)-сечений только с окрестностями вершин s и t является здесь случайным, приведем множество (00, 11)-сечений: $S_5(00, 11) = \{(01, 10, 03, 40), (01, 10, 12, 40), (01, 10, 03, 21), (01, 10, 12, 21)\} = \{(01 \wedge 10 \wedge ((03 \vee 12) \wedge (40 \vee 21)))\}$. На рис. 2 приведены подграфы исходного графа, полученные в результате этих сечений; вершины $s = 00$ и $t = 11$ помечены на рисунке концентрическими окружностями.

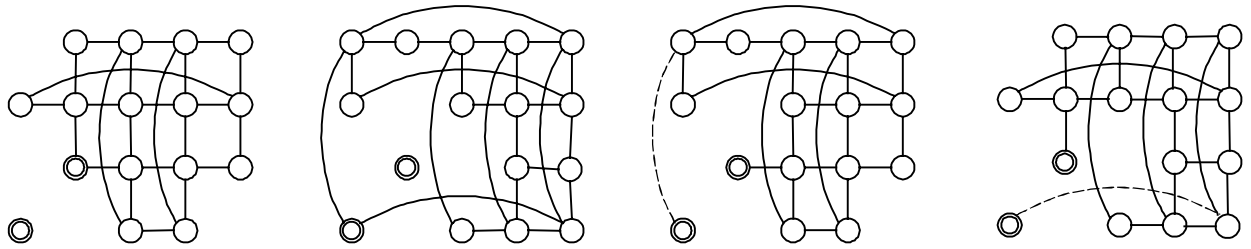

Рис. 2. d-ограниченные (00, 11)-сечения графа при $d = 5$

Проверив полученные в результате сечений компоненты связностей на их соответствие ограничению по диаметру $d \le 5$, выделим в них d-ограниченные компоненты связностей. Как было указано выше, для этого достаточно получить минимальные полные проекции [6] образовавшихся подграфов, в процессе построения которых и будут получены диаметры описанного каждой проекцией подграфа и их размеры. Нетрудно убедиться, что проекции $P_5'(11)$ и $P_5'(10)$ d-ограниченных компонент связности для сечений (01, 10, 03, 40) и (01, 10, 12, 21) совпадают с соответствующими компонентами $P'(11)$ и $P'(10)$, так как эти сечения изолируют только одну из концевых вершин. Размеры таких компонент $|V_5'| = V' = |\mathcal{N}[x]| = 15$, где $x = s \vee t$.

Ниже приведены проекции $P_6'(11)$ и $P_6''(11)$ компонент, образующихся в результате рассечения графа подмножествами вершин {01, 10, 12, 40} и {01, 10, 03, 21} соответственно:

$P_6'(11) = 11(_1 21(_2 20(_3 23(_4 13(_5 03(_6 00_6)_5), 22(_5 32_5), 33(_5 30, 32, 43_5)_4), 30(_4 31(_5 32, 41_5), 33(_5 23, 32, 43_5)_4)_3),$
$22(_3 23(_4 13(_5 03(_6 00_6)_5), 20(_5 30_5), 33(_5 30, 32, 43_5)_4), 32(_4 31(_5 30, 41_5), 33(_5 23, 30, 43_5), 42(_5 02, 41, 43_5)_4)_3),$
$31(_3 30(_4 20(_5 23_5), 33(_5 23, 32, 43_5)_4), 32(_4 22(_5 23_5), 33(_5 23, 30, 43_5), 42(_5 02, 41, 43_5)_4), 41(_4 42(_5 02, 32, 43_5)_4)_3)_2),$

$P_6''(11) = 11(_1 12(_2 02(_3 42(_4 32(_5 22, 31, 33_5), 41(_5 31, 40(_6 00_6)_5), 43(_5 33, 40(_6 00_6)_5)_4)_3),$
$13(_3 23(_4 20(_5 30_5), 22(_5 32_5), 33(_5 30, 32, 43_5)_4)_3),$
$22(_3 23(_4 13, 20(_5 30_5), 33(_5 30, 32, 43_5)_4), 32(_4 31(_5 30, 41_5), 33(_5 23, 30, 43_5), 42(_5 02, 41, 43_5)_4)_3)_2)_1).$

Данные проекции являются полными, так как 6-й уровень (выделенный в проекциях затенением) включает в себя последнюю из не определенных нижерасположенными уровнями вершин. Диаметры этих компонент одинаковы и равны 6, а число вершин в них $N' = 16$, изолированных вершин нет. Хотя подмножества удаленных вершин не являются здесь (00, 11)-сечениями в общеупотребительной терминологии, тем не менее удаление этих подмножеств приводит к утрате связанности некоторых вершин не только из-за их физической изолированности, но и из-за заданного ограничения достижимости, что соответствует введенному нами определению d-ограниченного (s, t)-сечения. Из рис. 2 видно, что хотя компонента связности включает в себя вершину 00, в состав проекций 5-ограниченных компонент связности графа, получаемых из $P_6'(11)$ и $P_6''(11)$ удалением 6-го уровня, эта вершина не входит. Ребра, увеличивающие диаметр компоненты связности до $d(G')$, показаны здесь штриховыми линиями.

Описанный выше подход позволяет выявлять не только минимальные d-ограниченные (s, t)-сечения, но и сечения большей мощности. При этом одновременно с задачей поиска множества сечений заданной мощности решается задача выявления компонент связности, их размеров и диаметров. Последовательное применение такого поиска на множестве пар вершин в графе позволяет выявить в множестве сечений заданной мощности эквивалентные с тем, чтобы использовать эту информацию в построении функций структурной отказоустойчивости и структурной живучести системы.

Заключение

Отказоустойчивость системы базируется на ее структурной конъюнктуре, заключающейся в наличии достаточного числа вершин в графе ВС и в обеспечении заданного критерия их связанности. В качестве последнего определено значение максимально допускаемого в системе эксцентриситета вершин, или диаметра компоненты связности графа ВС при ее деградации. Поставлена и решена задача выявления наименьшего числа l^* вершин, удаление которых переводит диаметр компоненты связности в закритическую область, или при сохранении диаметра в докритической области уменьшает число вершин в компоненте более чем на l^*.

Сформулированы используемые при поиске положения и предложен подход к решению проблемы, основанный на использовании формальных описаний структур ВС в виде скобочных образов и проекций соот-

ветствующих графов. Подход позволяет выявлять не только минимальные d-ограниченные (s, t)-сечения, но и сечения большей мощности. Одновременно с задачей поиска множества сечений заданной мощности решается задача выявления компонент связности, их размеров и диаметров. Последовательное применение такого поиска на множестве пар вершин в графе позволяет выявить в множестве сечений заданной мощности эквивалентные для построении функций структурной отказоустойчивости и структурной живучести системы. Параллельная алгоритмизация предложенного подхода сразу для множества (s, t)-сечений выглядит при этом вполне естественной; вполне допустимы и варианты поиска d-ограниченных сечений между подмножествами вершин в графе с использованием, к примеру, известного приема стягивания ребер внутри каждого из подмножеств.

ЛИТЕРАТУРА

1. *Мелентьев В.А.* Толерантность графов и структурная отказоустойчивость вычислительных систем // Вестник ТГУ. Приложение. 2004. № 9(I). С.144 – 150.
2. *Мелентьев В.А.* Функция структурной отказоустойчивости и d-ограниченная компонента связности графа вычислительной системы // Наст. сборник. С. 102 – 106.
3. *Мелентьев В.А.* Формальный подход к исследованию структур вычислительных систем // Вестник ТГУ. Приложение. 2005. № 14. С. 167 – 172.
4. *Мелентьев В.А.* Операции над проекциями графов и актуализация описаний отказоустойчивых систем // Вестник ТГУ. Приложение. 2006. № 17. С. 208 – 213.
5. *Харари Ф.* Теория графов. М.: Мир, 1973.
6. *Мелентьев В.А.* Формальные основы скобочных образов в теории графов // Труды II Междунар. конф. «Параллельные вычисления и задачи управления» PACO'2004. 2004. С. 694 – 706.
7. *Мелентьев В.А.* Изоморфизм графов и их образов в исследованиях отказоустойчивости систем // Вестник ТГУ. Приложение. 2005. № 14. С. 182 – 190.
8. *Мелентьев В.А.* Образ графа и поиск гамильтоновых путей // Вестник ТГУ. Приложение. 2005. № 14. С. 172 – 181.

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

DOI 10.17223/20710410/2/24

УДК 519.7

МИНИМИЗАЦИЯ ФУНКЦИОНАЛОВ, АССОЦИИРОВАННЫХ С ЗАДАЧАМИ КРИПТОГРАФИЧЕСКОГО АНАЛИЗА АСИММЕТРИЧНЫХ ШИФРОВ

В.И. Дулькейт, Р.Т. Файзуллин, И.Г. Хныкин

Омский государственный университет им. Ф.М. Достоевского

E-mail: r.t.faizullin@mail.ru

В работе рассматриваются численные методы минимизации функционалов, ассоциированных с задачами криптографического анализа асимметричных шифров. Показано, что для задачи факторизации подход позволяет получить строго более чем 50 % бит, определяющих ключ.

Ключевые слова: *криптографический анализ*, *алгоритм минимизации*, *ВЫПОЛНИМОСТЬ.*

Концепция

Пусть $L(x)$ – пропозициональная формула в конъюнктивной нормальной форме на множестве булевых переменных $x \in B^N$, где $B = \{0, 1\}$. Задача ВЫПОЛНИМОСТЬ (SAT) заключается в том, чтобы найти решающий набор $x_0 \in B^N$, такой, что $L(x_0) =$ ИСТИНА, или доказать, что решающего набора не существует.

Рассмотрим переход от задачи ВЫПОЛНИМОСТЬ к задаче поиска глобального минимума функционала вида (1).

Пусть дана КНФ:

$$L(x) = \bigcap_{i=1}^{M} c_i ,$$

где c_i – дизъюнкты вида $c_i = \bigcup\limits_{j<N} q_{i,j}(x_j)$. Здесь $q_{i,j}(x_j) = x_j$ или $\overline{x}_j$.

Сделаем переход к эквивалентной ДНФ:

$$L = \tilde{L}(x) = \bigcup_{i=1}^{M} \tilde{C}_i ,$$

где $\tilde{C}_i$ – конъюнкты вида $\tilde{C}_i = \bigcap\limits_{j<N} \tilde{Q}_{i,j}(x_j)$. Здесь $\tilde{Q}_{i,j}(x_j) = \overline{q}_{i,j}(x_j)$.

Рассмотрим функционал вида

$$\min_{x \in E^N} F(x) = \sum_{i=1}^{M} C_i(x), \text{ где } C_i(x) = \prod_{j=1}^{N} Q_{i,j}(x_j) .$$

Здесь

$$Q_{i,j}(x_j) = \begin{cases} (1-x_j)^2, & \text{если } x_j \in C_i(x), \\ x_j{}^2, & \text{если } \overline{x_j} \in C_i(x), \\ 1, & \text{иначе.} \end{cases} \qquad (1)$$

Суммирование ведется по всем M конъюнктам ДНФ, эквивалентной исходной КНФ. Соответствие между булевыми и вещественными переменными следующее: ЛОЖЬ→0, ИСТИНА→1.

Переход от булевой формуле к вещественной основан на использовании соответствия:

$$\begin{cases} y_i \vee y_j \to x_i + x_j, \\ y_i \wedge y_j \to x_i^2 x_j^2, \quad \text{где } \{y_i \in B, x_i \in R\}. \\ \overline{y}_i \to (1 - x_i), \end{cases}$$

Легко заметить, что $\min\limits_{x \in E^N} F(x) = 0$ соответствует достижению значения ИСТИНА на исходной КНФ.

Без потери общности можно рассмотреть 3-ДНФ, эквивалентную исходной КНФ:

$$J(x)=\sum_{\xi} z_i^2 z_j^2 z_k^2 \ , \text{ где } z_i=\begin{cases}1-x_i, & \text{если } x_i \in c_i(x),\\ x_i, & \text{если } \overline{x_i} \in c_i(x),\end{cases} \qquad c_i(x)-i \text{ триплет.} \tag{2}$$

Дифференцируя функционал по всем переменным x_i, получаем систему уравнений:

$$\sum_{\xi\in\Xi} z_j^2 z_k^2 x_i = \sum_{\xi\in\Lambda} z_j^2 z_k^2, \qquad i=1,2,...,P, \tag{3}$$

где

$$\Xi=\{\xi, i\in\xi : x_i \in c_i(x)\},$$
$$\Lambda=\left\{\xi, i\in\xi : \overline{x_i} \in c_i(x)\right\}$$

или $$A_i(x_1,...,x_{i-1},x_{i+1},...,x_n)\cdot x_i = B_i(x_1,...,x_{i-1},x_{i+1},...,x_n),\ i=1,...,P.$$

Коэффициенты A_i и B_i связаны соотношением $A_i(x_1,...,x_{i-1},x_{i+1},...,x_n) \ge B_i(x_1,...,x_{i-1},x_{i+1},...,x_n)$.

Поясним выбор представления исходной КНФ именно в виде эквивалентной 3-ДНФ. Дифференцируя функционал $F(x)$ (см. (1)) по всем переменным x_i, получаем систему уравнений, аналогичную (3), но количество вкладов в A_i и B_i определяется длиной скобок. Любая процедура решения этой системы при произвольной длине скобок будет естественным образом приводить к большим ошибкам округления. Ограничивая число переменных в скобках, мы исключаем эту техническую трудность.

Рассмотрим систему (3) как нелинейное операторное уравнение:

$$\Phi(x)=0\,. \tag{4}$$

Как показано в [5], применение метода Ньютона к решению данного уравнения неэффективно, так как решение принадлежит ядру производного оператора. Как альтернатива был предложен метод последовательных приближений с «инерцией»:

$$\left(\sum_{p=0}^{K}\sum_{\xi\in\Xi}\alpha_p x_i(t-p)^2 x_j(t-p)^2\right)x_k(t+1)=\sum_{\xi\in\Lambda} x_j^2(t)x_k^2(t) \sim A\cdot x_k(t+1)=B,$$
$$\sum_{p=1}^{K}\alpha_p=1, \quad \text{где } \alpha_p\in R[0,1]. \tag{5}$$

Имеется в виду то, что итерации происходят для вещественных чисел, а итоговый или промежуточный вектор проектируется на B^N, и уже на булевом векторе проверяется SAT. Ниже мы опишем различные модификации и гибридизации метода последовательных приближений с «инерцией» в применении к решению задачи K-SAT и покажем способы повышения эффективности алгоритма.

Гибридизация алгоритма

Основная процедура состоит из последовательных итераций, которые совмещают метод последовательных приближений и сдвиг по антиградиенту, так как правая часть (3) – это не что иное, как антиградиент исходного функционала.

Итерация состоит из двух блоков. Первый блок определяется формулой (5), используется схема Зейделя. Суть схемы Зейделя в том, что при нахождении очередного $x_i(t+1)$ на $(t+1)$-й итерации это значение подставляется вместо $x_i(t)$. Необходимо отметить, что реализация алгоритма допускает использование схемы Якоби, когда найденные $x_i(t+1)$ не используются в текущей итерации. Тесты показали, что схема Зейделя более устойчива в применении к решаемой задаче. Именно, после каждой итерации по схеме Зейделя значение функционала монотонно уменьшается, чего нельзя сказать о схеме Якоби. Кроме того, во всех случаях схема Зейделя быстрее приводила к решению. Отметим, что данное обстоятельство препятствует напрашивающейся простой схеме распараллеливания процесса решения с разделением данных.

Второй блок – реализация сдвига по градиенту. Рассмотрим (4). Пусть $\overline{x}(t)$ является решением, тогда $\Phi(\overline{x}(t))=0$. Уравнение (5) переписывается в виде $A(\overline{x}(t))\cdot\overline{x}(t)-B(\overline{x}(t))=0$. Это необходимое условие, которому должен удовлетворять вектор решения. Если текущее t-е приближение $\overline{x}(t)$ не является решением, то $A_i(\overline{x}(t))\cdot x_i(t)-B_i(\overline{x}(t))=p_i\neq 0$. Для итеративной формулы: $A_i(\overline{x}(t))\cdot x_i(t+1)-B_i(\overline{x}(t))=p_i$. Следовательно, чтобы удовлетворить необходимому условию, необходимо перейти к вектору:

$$\overline{x}_i(t+1)=\overline{x}_i(t)+p_i/A_i\,. \tag{6}$$

Очевидно, что после реализации (6) возможна ситуация, когда $\overline{x}_i(t+1)\notin R[0,1]$. В этом случае необходимо штрафным способом ограничивать $\overline{x}_i(t+1)$, иначе метод начинает экспоненциально расходиться. Осо-

бенно это проявляется на K-SAT формулах при $K > 4$. При приближении к решению скорость сходимости может сильно уменьшаться, т.е. алгоритм формирует цикл, лежащий на некотором плато (поверхность, определяемая функционалом), и траектория, образованная последовательными приближениями, более не выходит за пределы этого плато. Чтобы сойти с плато и продолжить сходимость к решению, применяется т.н. метод смены траектории.

Метод смены траектории заключается в поиске нового вектора приближения, который бы обладал свойствами, не худшими, чем текущий вектор приближения, но позволял бы продолжить поиск решения. Суть метода нахождения такого вектора в следующем.

Рассмотрим 3-КНФ, эквивалентную исходной 3-ДНФ (1):

$$K(x) = \prod_{\xi} (z_i + z_j + z_k), \text{ где } z_i = \begin{cases} x_i, & \text{если } x_i \in c_i(x), \\ 1 - x_i, & \text{если } \overline{x_i} \in c_i(x), \end{cases} \quad c_i(x) - i \text{ триплет}, \tag{7}$$

$$K(x) = 1 \Leftrightarrow (z_i + z_j + z_k) \neq 0, \forall \xi .$$

Для данного приближения $\bar{x}$ рассмотрим множество переменных:

$$E0 = \{ x_i \mid \exists \text{ триплет } c_i : x_i \text{или} \neg x_i \in c_i \ \& \ c_i(\bar{x}) = 0 \} .$$

С вероятностью *m_p* поменяем значения x_i на противоположные. При этом вероятность того, что другие триплеты станут невыполнимыми, невысока. Экспериментально установлено, что полученный вектор x_0 обладает свойствами, не худшими, чем $\bar{x}$ (количество невыполнимых триплетов до и после операции примерно одинаково). Используя данный вектор x_0 в качестве нового начального приближения, алгоритм очень быстро (в большинстве случаев за 5 – 10 итераций) находит следующее приближение, на котором функционал $F(x)$ достигает значения не хуже, чем на векторе $\bar{x}$. При этом, очень часто, удается проскочить плато, но при дальнейшем движении по новой траектории метод может зациклиться на другом плато. Тогда метод смены траектории повторяется.

Дополнительно рассмотрим множество:

$$E1 = \{ x_i \mid \exists \text{ триплет } c_i : x_i \text{или} \neg x_i \in c_i \ \& \ c_i(\bar{x}) = 1 \} .$$

Введем вероятности *m_p*0, *m_p*1. С вероятностью *m_p*0 при смене траектории будем использовать множество *E*0. С вероятностью *m_p*1 – множество *E*1. Вероятность *m_p*0 влияет на величину изменения вектора и при этом количество невыполнимых триплетов не увеличивается. Вероятность *m_p*1 влияет на качественное изменение вектора, количество невыполнимых триплетов может увеличиться. В принципе, чем выше *m_p*1, тем меньшую роль играют рестарты. Метод смены траектории применяется при достижении условия $|F(\bar{x}_2) - F(\bar{x}_1)| < \varepsilon_2$.

Преобразование исходной КНФ методом резолюции

Данное преобразование позволяет получить КНФ с меньшим количеством дизъюнктов и литералов, эквивалентную исходной. «Резольвента» – дизъюнкция конъюнктов, отличающихся знаком по единственной переменной. Все возможные резольвенты добавляются к КНФ и используются для вычисления других резольвент.

Дублирующие конъюнкты и тавтологии удаляются, и используется сокращенная процедура с глубиной рекурсии 1. Вычислительная сложность процедуры $O(n \cdot \log n)$. Метод резолюции в применении к КНФ, ассоциированных с задачами факторизации и дискретного логарифмирования (см. ниже), позволяет уменьшить исходное число конъюнктов до 50 % и иногда разрешить до 20 % переменных. Подробно о методе резолюций можно найти в [6]

Результаты численных экспериментов

После каждой модификации проводилось тестирование алгоритма для определения эффективности проделанных изменений. При тестировании использовалось несколько типов примеров: тесты с соревнований решателей SAT 2005 года www.lri.fr/~simon/, тесты библиотеки SATLib www.cs.ubc.ca/~hoos/, тесты, сформированные для задач факторизации и дискретного логарифмирования, тесты для КНФ больших размерностей, сформированные случайным образом. Подробные результаты представлены в [5].

Оказалось, что сдвиг по градиенту хорошо сокращает погрешности и ускоряет сходимость алгоритма. Вычислительные эксперименты со случайными формулами показали заметное уменьшение времени решения тестов. Число решенных примеров увеличилось примерно на 20 %. Применение данного приема позволило достаточно эффективно решать тесты uf20-91 (из 1000 тестов решены 703). Однако некоторые тесты решались только после задания определенного начального приближения, что говорит о необходимости рестартов. На примерах uf250-1065 алгоритм показал результат 6 % от стандартных трудных тестов (предыду-

щая версия алгоритма – 1 %). Тесты SAT-2005 (OKGenerator10000-42000 – 10000 переменных, 42000 скобок) использовались в сокращенной форме. Максимально удалось решить подформулы из 36000 скобок (предыдущая версия алгоритма – 35000 скобок). На подформулах из более чем 36000 скобок метод зацикливается на некотором плато, что говорит о необходимости смены траектории.

Метод смены траектории существенно увеличил число решаемых примеров. Результаты представлены в табл. 1.

Таблица 1

Результаты тестирования алгоритма + метод смены траектории

Наименование теста	Количество литералов (N)	Количество дизъюнктов (M)	Число тестов	% решенных тестов	Максимальное число итераций
Backbone-minimal Sub-instances (*формулы с минимальным хребтом*), *3-SAT*					
RTI	100	429	500	98,6	19988
BMS	100	<429	500	79,8	29831
Controlled Backbone Size Instances (*b – размер хребта*), *3-SAT*					
CBS_b10	100	403	1000	100	38972
CBS_b10	100	449	1000	100	38880
CBS_b90	100	449	1000	98	29738
Uniform Random 3-SAT (*UF*)					
uf20-91	20	91	1000	100	448
uf250-1065	250	1065	100	98	9731
SAT-encoded "Flat" Graph Colouring Problems					
flat30-60	90	300	100	100	4317

Увеличение разрядности вычислений

Была исследована сходимость алгоритма при увеличении разрядности вычислений. Испытания с типами DOUBLE и FLOAT показали преимущество вычислений с двойной точностью. При переходе на тип DOUBLE количество решенных примеров увеличивается на 10 %, скорость сходимости в среднем также увеличивается. Дальнейшее увеличение разрядности к значимому эффекту не приводит.

Возможные практические применения

В последнее время наблюдается повышенный интерес к проблеме кодирования криптографических алгоритмов в терминах задачи выполнимости (SAT). В работе [4] эскизно иллюстрируется подход, позволяющий в принципе свести задачу факторизации к SAT. Целью данного исследования является построение алгоритмов генерации эквивалентных, но различных КНФ и последующей минимизации ассоциированного функционала для задач факторизации, дискретного логарифмирования и дискретного логарифмирования на эллиптической кривой.

Результаты работы алгоритма для задачи факторизации

Результаты приведены в табл. 2. Для группы задач длины до 72 бит факторизуемого числа были получены точные решения. При этом эффективность предложенного метода превосходит известные нам алгоритмы.

Таблица 2

Результаты тестирования полного алгоритма для задачи факторизации

Число бит в факторизуемом числе	Количество литералов	Количество дизъюнктов	Время решения методом последовательных приближений	Время решения алгоритмом RANOV (победитель 2005 г.)	Время решения алгоритмом SATz (один из лучших переборных алгоритмов)
20	254	4979	0,1 с	0,1 с	0,1 с
32	801	17867	2 мин	>1 ч	1,8 мин
40	990	22333	7 мин	>1 ч	12 мин
44	1199	27291	36 мин	>1 ч	>1 ч
48	1428	32741	3,5 ч	>10 ч	>10 ч
56	1946	45141	36 мин	>10 ч	>10 ч
60	2235	52079	10,2 ч	>20 ч	>20 ч
68	2873	67455	79 ч	>100 ч	>100 ч
72	3222	75881	168 ч	>200 ч	>200 ч

Другой интересный результат в том, что уже после первых нескольких сотен итераций метод находит более 59 % верных бит решения. Трудность заключается в распознавании, какие именно биты были опреде-

лены верно. При этом при росте числа бит исходного факторизуемого числа происходит рост процентного отношения числа верных бит для соотношения, определяющего факторизуемое число (рис. 1).

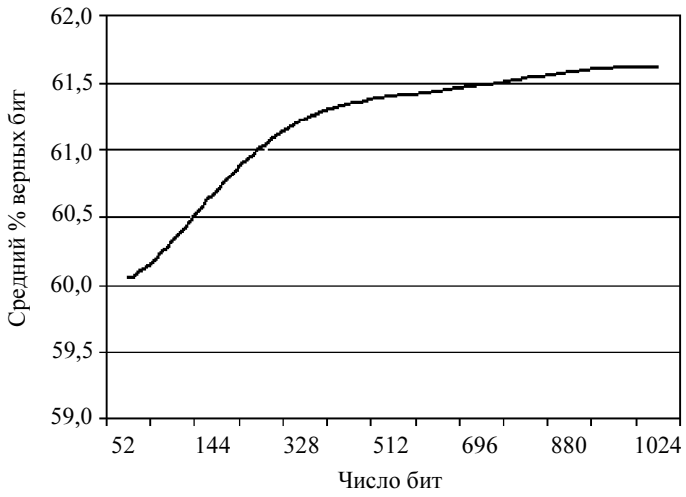

Рис. 1. Рост процентного отношения числа верных бит
к числу бит факторизуемого числа

Для тестирования использовалось 50 различных тестов для каждой размерности. В качестве сомножителей выбирались числа, удовлетворяющие всем тестам, гарантирующим криптостойкость RSA. На рис. 1 представлены усредненные данные. Обратим внимание на то, что вне зависимости от размерности задачи для каждого теста проводилось всего 1000 итераций. Исходя из рисунка, можно сделать предположение о том, что рост числа верных бит в зависимости от длины факторизуемого числа имеет логарифмический характер.

Сведение к КНФ задачи факторизации

Рассмотрим непосредственно алгоритм сведения к КНФ задачи факторизации. Требуется для заданного числа n получить КНФ, решающий набор которой существует тогда и только тогда, когда n – составное число. Кроме того, решающий набор должен содержать все биты двоичного представления нетривиальных делителей n. Без потери общности рассмотрим классический алгоритм умножения «столбиком». Будем отождествлять биты сомножителей и результата с литералами (свободными логическими переменными). Результат умножения первого сомножителя на i-й бит второго можно представить в виде вектора $\overline{P_i}$. Именно суммирование всех этих векторов представляет основную сложность. Поэтому предлагается выполнять эту операцию последовательно с сохранением результата в промежуточных векторах $\overline{S_k}$.

Весь процесс вычисления можно разбить на три этапа относительно операции сложения:

1. Сложение векторов, составленных из произведений двух литералов. Выполняется один раз. В результате этой операции будет заполнен вспомогательный вектор сумм $\overline{S_1}$ и вычислено два младших бита результата. Условно данный этап можно записать так: $\overline{P_1} + \overline{P_2} = (\overline{S_2}, r_2, r_1)$.

2. Суммирование вектора $\overline{S_k}$ с вектором произведений. Выполняется $N-3$ раз. В результате заполняется массив $\overline{S_{k+1}}$ и вычисляется очередной бит результата $\overline{S}_{i-2} + \overline{P}_i = (\overline{S}_{i-1}, r),\ i = 3...N-1$.

3. Последнее суммирование вектора $\overline{S_k}$ с вектором произведений. Выполняется один раз. В результате вычисляются оставшиеся биты результата $\overline{S}_{N-2} + \overline{P}_N = (r_{2N}...r_N)$.

Теперь перейдём к рассмотрению идеи генерации КНФ. Простейший случай – приравнивание одного литерала другому: $x = y$. Данное равенство будет справедливо тогда и только тогда, когда истинна формула $(\overline{x} \vee y)(x \vee \overline{y})$.

Другим часто встречающимся выражением является

$$x = A \oplus B \oplus C. \tag{8}$$

Поступая аналогичным образом, получаем эквивалентную формулу:

$$\left(\overline{x} \vee (A \oplus B \oplus C)\right)\left(x \vee \overline{(A \oplus B \oplus C)}\right) = \left(Ax \oplus Bx \oplus Cx \oplus \overline{x}\right)\left(A\overline{x} \oplus B\overline{x} \oplus C\overline{x}\right). \tag{9}$$

Для представления правой части в виде КНФ воспользуемся леммами 1 и 2.

Лемма 1.

$$\bigoplus_{i=1}^{N} x_i = \prod_{\{\delta_i\}\in M_N} \left(x_1^{\delta_1} \vee x_2^{\delta_{21}} \vee \cdots \vee x_N^{\delta_N} \right),$$

где в левой части сумма по модулю 2, M_N – множество двоичных векторов длины N, содержащих чётное число нулей. Операция «возведения в степень» имеет стандартный для булевой алгебры смысл:

$$x^{\delta} = \begin{cases} \overline{x}, & \delta = 0, \\ x, & \delta = 1. \end{cases}$$

Лемма 2.

$$\bigvee_{i=1}^{N} x_i^{\delta_i} \vee \prod_{j=1}^{L} y_j^{\sigma_j} = \prod_{\{\pi_k\}\in 2^L/\{0,0,\ldots,0\}} \left(\bigvee_{i=1}^{N} x_i^{\delta_i} \vee \bigvee_{j=1}^{L} \left(y_j^{\sigma_j} \right)^{\pi_k} \right).$$

После применения леммы 1 будут получены конъюнкты следующего вида:

$$\left(d \vee \overline{\bar{a}\bar{b}c} \vee x\bar{y}c \right),$$

т.е. можно выделить 3 вида дизъюнктов внутри каждого конъюнкта:

1. Одиночные литералы (то, к чему следует стремиться: в правильной КНФ все дизъюнкты должны быть одиночными литералами).

2. Дизъюнкты вида $\overline{\prod x_i^{\delta_i}}$, которые по правилу де Моргана можно легко свести к одиночным литералам.

3. Дизъюнкты вида $\prod x_i^{\delta_i}$, наиболее трудный случай, сведение скобок с такими дизъюнктами к КНФ иллюстрируется леммой 2.

Отдельного рассмотрения заслуживает операция вычисления переноса в следующий разряд при суммировании трёх слагаемых. Перенос может быть вычислен через соответствующую сумму:

$$c = \mathrm{carry}(x, y, z, \mathrm{sum}) = \overline{\left(\mathrm{sum} \oplus xyz \oplus \overline{\bar{x}\bar{y}\bar{z}} \right)}.$$

Выполнение данного равенства эквивалентно истинности следующей формулы:

$$\left(\bar{c} \vee \overline{\left(\mathrm{sum} \oplus xyz \oplus \overline{\bar{x}\bar{y}\bar{z}} \right)} \right) \cdot \left(c \vee \left(\mathrm{sum} \oplus xyz \oplus \overline{\bar{x}\bar{y}\bar{z}} \right) \right).$$

Приведённое выражение можно преобразовать, положив

$$x = \bar{c},\ A = \mathrm{sum},\ B = xyz,\ C = \overline{\bar{x}\bar{y}\bar{z}}.$$

И далее описанной выше процедурой можно построить соответствующую КНФ. Трудоемкость полученного алгоритма оценивается как $O(n^2)$ в зависимости от количества бит исходного числа. Для факторизации числа, представляемого двоичным вектором длиной 1024 бит, получились КНФ с 500 000 переменными, 12 000 000 скобок. Отметим, что результат о превышении числа верных бит после нескольких тысяч итераций алгоритма (5) относится именно к числу переменных. Например, из 500 000 переменных мы получаем в среднем более чем 300 000 верных, что в силу очевидных соотношений между битами переноса по строке, отвечающей нулевому биту, может существенно облегчить решение основной задачи.

Сведение к КНФ других задач криптоанализа

Были получены аналогичные алгоритмы сведения для задач дискретного логарифмирования и дискретного логарифмирования на эллиптической кривой. Предложен алгоритм генерации множества эквивалентных КНФ для задачи факторизации, учитывающий неделимость на малые простые числа. Последнее обстоятельство позволяет строить параллельные версии приведенных выше алгоритмов и методикой «голосования бит» определять верные биты с определенной вероятностью.

Заключение

Разработанный метод не уступает известным методам решения SAT на многих группах тестовых примеров и превосходит их на тестах задачи факторизации больших размерностей. Показан рост относительного числа верно найденных бит для задачи факторизации с ростом размерности задачи. Так, для рабочего числа бит 1024 среднее значение верных бит равно 61,75 %, что больше стартового значения в 59,5 % для 50 бит. Кроме того, показано наличие «слабых» размерностей, для которых метод является эффективным. Анало-

гичные результаты получены и для задачи дискретного логарифмирования, но основным препятствием здесь является высокая размерность получающихся задач. Так, для 1024 бит число переменных в функционале оценивается величиной 1000 000 000, а число дизъюнктов на порядок больше. Результаты точного решения КНФ, эквивалентных задаче дискретного логарифмирования, приведены в табл. 3.

Таблица 3

Результаты тестирования полного алгоритма для задачи дискретного логарифмирования

Размерность, бит	Количество литералов	Количество дизъюнктов	Время решения методом последовательных приближений, с	Время решения алгоритмом RANOV, с (победитель 2005 г.)	Время решения алгоритмом SATz, с (один из лучших переборных алгоритмов)
18	28224	448018	63,57	97,23	81,16
20	38840	623239	108,20	>1800	>1800
22	51832	839032	182,73	>1800	>1800
24	67440	1099630	277,46	>1800	>1800
26	85904	1409250	417,71	>1800	>1800

Примечание. Знак «>» означает, что за указанное время решение найдено не было.

ЛИТЕРАТУРА

1. *Беспалов Д.В.* Пропозициональные алгоритмы в задачах несимметричной криптографии // Вестник ТГУ. Приложение. 2004. № 9(I).
2. *Беспалов Д.В.*, *Семёнов А.А.* О логических выражениях для задачи 2-ФАКТОРИЗАЦИЯ // Вычислительные технологии. 2002. Т. 7. Ч. 2.
3. *Найфе А.* Методы возмущений. М.: Мир, 1976.
4. *Файзуллин Р.Т.* О решении нелинейных алгебраических систем гидравлики // Сибирский журнал индустриальной математики. 1999. № 2. С. 176 – 184.
5. *Файзуллин Р.Т.*, *Хныкин И.Г.*, *Дулькейт В.И.*, *Салаев Е.В.* Алгоритм минимизации функционала, ассоциированного с задачей 3-SAT и его практические применения. Челябинск, 2007.
6. *Хныкин И.Г.* Модификация КНФ, эквивалентных задачам криптоанализа асимметричных шифров, методом резолюции // ИТМУ. 2007. № 8.
7. *Cook S.A.* The Complexity of Theorem Proving Procedures. Proceedings Third Annual ACM Symposium on Theory of Computing, May 1971.
8. *Gu J.*, *Purdom P.W.*, *Franco J.*, *Wah B.W.* Algorithms for the Satisfiability Problem: A Survey // DIMACS Series in Discrete Mathematics and Theoretical Computer Science. 1996. P. 19 – 151.

2008 Вычислительные методы в дискретной математике № 2(2)
DOI 10.17223/20710410/2/25
УДК 519.7

АНАЛИЗ НЕКОТОРЫХ КРИПТОГРАФИЧЕСКИХ ПРИМИТИВОВ НА ВЫЧИСЛИТЕЛЬНЫХ КЛАСТЕРАХ[1]

А.А. Семенов, О.С. Заикин, Д.В. Беспалов, П.С. Буров, А.Е. Хмельнов

Институт динамики систем и теории управления СО РАН, г. Иркутск

E-mail: biclop@rambler.ru, oleg.zaikin@icc.ru, bespalov@altrixsoft.com, evle.zzz@gmail.com, alex@icc.ru

Работа является продолжением серии статей, посвященных обращению дискретных функций из класса, имеющего пересечения с различными областями кибернетики. В данный класс, в частности, попадают функции, используемые в современных криптосистемах в качестве шифрующих преобразований. В настоящей работе обсуждаются некоторые характерные моменты, касающиеся решения задач обращения рассматриваемых функций на многопроцессорных вычислительных системах.

Ключевые слова: *обращение дискретных функций*, *преобразования Цейтина*, *многопроцессорный кластер.*

Данная работа предполагает освещение имеющихся результатов и перспектив дальнейших исследований в решении булевых уравнений большой размерности с приложениями в задачах криптографии. Особое внимание уделяется использованию для решения рассматриваемых классов логических уравнений многопроцессорных вычислительных систем (кластеров).

На сегодняшний день основу программ, наиболее успешно работающих с логическими уравнениями, составляют алгоритмы решения SAT-задач. Тем самым один из доминирующих на данный момент подходов состоит в сведении исходной (разнородной) системы логических уравнений к уравнению вида

$$C(x_1, \ldots, x_n) = 1, \quad (1)$$

в левой части которого находится конъюнктивная нормальная форма (КНФ) над множеством булевых переменных $X = \{x_1, \ldots, x_n\}$.

В п. 1 работы предполагается дать краткий обзор техники преобразований Цейтина [1] – основного инструмента, позволяющего сводить системы логических уравнений к уравнениям в формате (1); п. 2 содержит краткое описание программного комплекса «Transalg», реализация в рамках которого преобразований Цейтина позволяет сводить к уравнениям вида (1) задачи обращения дискретных функций из широкого класса (данный класс включает в себя большинство функций, используемых в системах шифрования).

В п. 3 описывается технология крупноблочного распараллеливания SAT-задач для их последующего решения на вычислительных кластерах. Данная технология была с успехом применена в криптоанализе ряда генераторов ключевого потока [2, 3]. Здесь же отмечено, что используемая концепция крупноблочного параллелизма применима не только к SAT-задачам, но и к многочисленным классам задач с «булевыми данными».

В п. 4 подробно описан программный комплекс D-SAT – основной инструмент в решении задач обращения дискретных функций на вычислительных кластерах.

В п. 5 описывается технология построения прогнозов трудоемкости задач криптоанализа (в форме SAT-задач) при их решении на вычислительных кластерах. В основе данной техники лежит идеология крупноблочного параллелизма, позволяющая строить прогнозы параллельного времени решения SAT-задач при варьируемых значениях ряда входных параметров. В качестве иллюстрирующего примера рассматривается логический криптоанализ широко известной системы поточного шифрования A5/1.

1. Преобразования Цейтина логических уравнений

Преобразования Цейтина в применении к логическим уравнениям реализуют фундаментальную идею, которую можно назвать «концепцией подстановки». Данная идея, состоящая в замене некоторой части формулы новой переменной (с последующими дополнительными преобразованиями), возникает в различных областях математики. В контексте логических уравнений такого рода преобразования впервые были использованы Г.С. Цейтиным в 1968 г. в работе [1]. Данную статью следует считать, по-видимому, наиболее ранней работой, в которой анализировались вопросы сложности вывода в исчислении высказываний. В ней, в частности, было установлено, что совмещение стратегии общей резолюции (в смысле Дж.А. Робинсона, [4])

[1] Работа выполнена при поддержке гранта РФФИ № 07-01-00400-а и при частичной поддержке гранта Президента РФ НШ–1676.2008.1.

с преобразованиями Цейтина[1] позволяет на некоторых классах логических противоречий получить «экспоненциальный выигрыш» по сложности в сравнении с регулярной резолюцией. Статья [1] была перепечатана на английском языке в 1983 г. [6].

В несколько более общем (по сравнению с [1]) контексте преобразования Цейтина были использованы в [7] при формулировке основных концепций логического криптоанализа[2]. Примерно в то же время [8] было показано, что преобразования Цейтина обладают свойством консервативности в смысле [9] (см. также [10]). Этот факт очень важен по отношению именно к задаче обращения дискретных функций, поскольку он позволяет гарантировать ее корректное решение.

Напомним, что логическими уравнениями называются выражения вида

$$L(x_1, \ldots, x_n) = 0, L(x_1, \ldots, x_n) = 1,$$

где $L(x_1, \ldots, x_n)$ – формула исчисления высказываний над множеством булевых переменных $X = \{x_1, \ldots, x_n\}$. Поиск решения логического уравнения

$$L(x_1, \ldots, x_n) = \beta, \beta \in \{0,1\},$$

означает поиск такого набора $(\alpha_1, \ldots, \alpha_n)$, $\alpha_i \in \{0, 1\}$, $i \in \{1, \ldots, n\}$, что $L(x_1,...,x_n)|_{(\alpha_1,...,\alpha_n)} = \beta$. Здесь через $L(x_1,...,x_n)|_{(\alpha_1,...,\alpha_n)}$ обозначена подстановка в $L(x_1, \ldots, x_n)$ соответствующих значений переменных: $x_1 = \alpha_1$, …, $x_n = \alpha_n$. Если такого набора не существует, то рассматриваемое логическое уравнение не имеет решений.

Будем рассматривать задачу поиска решений логических уравнений в следующей общей постановке:

$$F\left(h_1\left(x_1^1,...,x_{r_1}^1\right),...,h_s\left(x_1^s,...,x_{r_s}^s\right)\right) = 1. \quad (2)$$

Здесь $h_1, \ldots, h_s$ – некоторые (в общем случае сложные) булевы функции. Все булевы функции здесь и далее полагаются всюду определенными. Положим

$$X = \{x_1,...,x_n\} = \bigcup_{i=1}^{s}\left\{x_1^i,...,x_{r_i}^i\right\},$$

таким образом, F: $\{0, 1\}^n \to \{0, 1\}$. Введем булеву функцию

$$g_{h_1}\left(x_1^1,...,x_{r_1}^1,u_1\right), g_{h_1} : \{0,1\}^{r_1+1} \to \{0,1\}:$$

$$g_{h_1}\left(x_1^1,...,x_{r_1}^1,u_1\right) = \begin{cases} 1, h_1\left(x_1^1,...,x_{r_1}^1\right) = u_1, \\ 0, h_1\left(x_1^1,...,x_{r_1}^1\right) \neq u_1. \end{cases}$$

Рассмотрим логическое уравнение

$$C\left(g_{h_1}\left(x_1^1,...,x_{r_1}^1,u_1\right)\right) \cdot F\left(u_1,...,h_s\left(x_1^s,...,x_{r_s}^s\right)\right) = 1. \quad (3)$$

Здесь $C\left(g_{h_1}\left(x_1^1,...,x_{r_1}^1,u_1\right)\right)$ – КНФ-представление булевой функции g_{h_1} над $\left\{x_1^1,...,x_{r_1}^1,u_1\right\}$. Переход от уравнения (2) к уравнению (3) – это одна итерация преобразований Цейтина.

Теорема 1 (см. [11]). Существует взаимно однозначное соответствие между множествами решений уравнений (2) и (3). Переход от произвольного решения уравнения (3) к соответствующему решению уравнения (2) осуществляется за линейное время.

Легко заметить, что, применив преобразования Цейтина к уравнению (2) полиномиально ограниченное от объема кодировки (2) число раз, можно перейти от уравнения (2) к уравнению вида (1), причем множество его решений будет равномощно множеству решений (2). Более того, вместо КНФ-представлений функций g можно использовать их представления в виде любых «нормальных форм».

2. Пропозициональное представление алгоритмов вычисления дискретных функций; программный комплекс «Transalg»

Преобразования Цейтина являются базой «пропозиционального подхода» к задаче обращения дискретных функций одного важного класса.

Под дискретными функциями здесь понимаются функции вида f_n: $\{0, 1\}^n \to \{0, 1\}^*$. Если натуральное семейство $\{f_n\}_{n \in N}$ вычисляется при помощи фиксированной программы M машины Тьюринга, то данное семейство функций называется алгоритмически вычислимым. Если при этом каждая функция f_n семейства определена всюду на $\{0, 1\}^n$, то говорят, что данное семейство образовано всюду определенными алгоритмически вычислимыми функциями. Для таких функций стандартным образом (см., например, [10]) вводится понятие алгоритмической сложности. Все возможные всюду определенные алгоритмически вычислимые за полиномиальное время дискретные функции образуют класс $\Im$. Очевидно, что данный класс включает в себя, в частности, все функции, используемые в криптосистемах в качестве шифрующих преобразований. За-

[1] Данный тип вывода получил название «расширенная резолюция» (см., например, [5]).

[2] Авторы на тот момент еще не знали, что фактически они используют преобразования Цейтина.

дача обращения произвольной функции $f_n \in \Im$ состоит в следующем: по произвольному $y \in \mathrm{Ran}\, f_n$ найти такой $x \in \mathrm{Dom}\, f_n = \{0, 1\}^n$, что $f_n(x) = y$ (через $\mathrm{Dom}\, f_n$ и $\mathrm{Ran}\, f_n$ обозначены соответственно область определения и область значений функции f_n).

В работах [7, 11, 2, 3] последовательно развивался подход к обращению функций класса $\Im$, базирующийся на идее пропозиционального представления прямых алгоритмов их вычисления. Данная концепция в своей основе восходит к работе С. Кука [12], однако ее практическое воплощение требует многочисленных детализаций.

Первая программная реализация «пропозиционального подхода» была осуществлена в [13]. Созданный программный комплекс (LC-комплекс) с успехом применялся в криптоанализе различных систем поточного шифрования. Основу LC-комплекса составлял LC-язык, представляющий собой фактически фрагмент языка С. Алгоритмы, записываемые на LC-языке, транслировались в логические выражения в формате КНФ при помощи специального транслятора (LC-транслятор). LC-комплекс оказался удобен для сведения задач криптоанализа блочных шифров к SAT-задачам. Однако для логического криптоанализа систем шифрования, в алгоритмах которых используются условные переходы, возможностей LC-комплекса было недостаточно. На сегодняшний день данный комплекс не удовлетворяет многообразию задач, вписывающихся в концепцию пропозиционального подхода к обращению функций класса $\Im$.

Все перечисленные причины привели к выводу об актуальности создания нового программного комплекса для пропозиционального представления алгоритмов. Данный проект получил название «Transalg» [14]. В настоящий момент проект находится в стадии активной разработки. Комплекс «Transalg» имеет модульную архитектуру и представляет собой среду, в которую интегрированы компоненты, осуществляющие трансляцию алгоритмов, преобразования форматов, решение логических уравнений и проверку получаемых решений. Модульность архитектуры Transalg'а позволяет задействовать для решения логических уравнений различные методы (SAT-подход, BDD-подход, работа с системами полиномиальных уравнений над GF(2) и др.).

3. Крупноблочный параллелизм в булевых задачах; прогнозирование параметров распараллеливания

Кратко опишем технологию крупноблочного распараллеливания, использованную в решении SAT-задач и апробированную на задачах логического криптоанализа некоторых генераторов двоичных последовательностей [2, 3]. В [15] показано, что основные идеи [2] переносятся без существенного изменения на разнообразные задачи с «булевыми данными». В изложении данного материала в основном будем следовать работе [15].

Отметим, что различные задачи, использующие двоичные данные, можно объединить в контексте общей постановки «удовлетворения ограничениям». А именно, будем рассматривать задачи следующего типа. Дан набор формул-условий вида

$$\Phi(x_{i_1^1},...,x_{i_{r_1}^1}) \rhd A_1,...,\Phi(x_{i_1^m},...,x_{i_{r_m}^m}) \rhd A_m, \tag{4}$$

где $\bigcup_{j=1}^{m}\{x_{i_1^j},...,x_{i_{r_j}^j}\} = \{x_1,...,x_n\} = X$ – множество переменных, принимающих значения в $\{0, 1\}$, $\rhd \in \{=,\geq\}$, $A_j, j \in \{1, \ldots, m\}$ – элементы GF(2) или целые числа. Формулы (4) – это выражения вида «КНФ = 1», системы линейных неравенств вида

$$A \cdot x \geq b,$$

(где A – матрица размерности $m \times n$ с элементами из множества $\{-1, 0, 1\} \cap Z$; b – целочисленный вектор размерности m), либо полиномиальные уравнения над полем GF(2). Требуется найти вектор значений истинности переменных из X, удовлетворяющий всем условиям (4). Такой вектор будем называть решением системы (4).

Для произвольного множества $X^d = \{x_{i_1},...,x_{i_d}\}$, $\{i_1, \ldots, i_d\} \subseteq \{1, \ldots, n\}$, определим подстановку набора значений переменных

$$x_{i_1} = \alpha_{i_1},...,x_{i_d} = \alpha_{i_d}, \alpha_{i_j} \in \{0,1\}, j \in \{1,...,d\}, \tag{5}$$

в произвольную формулу вида (4) как замену вхождения переменной $x_{i_j}, j \in \{1,...,d\}$, константой α_{i_j} с последующим выполнением соответствующих преобразований (когда это возможно). Преобразования, являющиеся результатами применения подстановок типа (5) к выражениям вида (4), – это вычеркивания соответствующих литералов и дизъюнктов из КНФ, перенос целых чисел (с противоположным знаком) в правую часть линейных неравенств, а также вычеркивание мономов из булевых полиномов либо замена некоторых мономов на константу «1» с последующим сложением по mod 2 нескольких констант. Обозначим через 2^{X^d} множество, образованное всевозможными двоичными векторами длины d, каждый из которых определяет

значения истинности переменных из X^d. Каждому вектору $\alpha = (\alpha_1, ..., \alpha_d) \in 2^{X^d}$ поставим в соответствие результат следующей подстановки в выражения (4): $x_{i_1} = \alpha_1, ..., x_{i_d} = \alpha_d$. Тем самым вектору $\alpha = (\alpha_1, ..., \alpha_d) \in 2^{X^d}$ сопоставляется набор преобразованных выражений типа (4), который будем обозначать через Φ^α. Таким образом, множество X^d задает естественное соответствие между набором формул вида (4) и семейством формул $\{\Phi^\alpha\}_{\alpha \in 2^{X^d}}$. Каждая формула данного семейства есть либо логическое уравнение вида «КНФ = 1», либо система линейных неравенств вида (2), либо система полиномиальных уравнений над полем GF(2). Будем также говорить, что множество X^d задает декомпозицию задачи решения системы (4) на семейство задач поиска решений систем $\{\Phi^\alpha\}_{\alpha \in 2^{X^d}}$. Дополнительно полагаем, что при $d = 0$ результатом такой декомпозиции является исходная система типа (4).

Несложно видеть, что если двоичный вектор β – решение системы Φ^α при фиксированном $\alpha \in 2^{X^d}$, то компоненты векторов α и β дают компоненты некоторого решения исходной системы (4). Наоборот, если γ – произвольное решение (4), то часть его компонент определяет некоторый вектор $\alpha \in 2^{X^d}$, а оставшиеся компоненты определяют вектор β, являющийся решением системы Φ^α. Таким образом, задача поиска решения системы (4) сводится к задаче поиска решений систем Φ^α, $\alpha \in 2^{X^d}$, причем (4) не имеет решений (несовместна) тогда и только тогда, когда при всех $\alpha \in 2^{X^d}$ системы Φ^α несовместны.

Задачи поиска решений систем вида Φ^α – это, по сути, частные случаи исходной задачи, имеющие меньшую размерность, и для их решения можно использовать многопроцессорный вычислительный кластер. Более точно, упорядоченное некоторым образом семейство систем $\{\Phi^\alpha\}_{\alpha \in 2^{X^d}}$ может рассматриваться как очередь заданий для *r*-процессорного кластера. Именно такой подход был использован в [2] применительно к решению SAT-задач.

В этой же работе была предложена процедура статистического прогнозирования размерности множества X^d, то есть значения параметра *d*, при котором суммарное время решения исходной задачи на вычислительном кластере минимально. Здесь мы переносим данный результат работы [2] на общую задачу поиска решений систем вида (4).

Предположим, что рассматривается некоторая задача типа (4). Через *S* обозначаем решатель, используемый для поиска ее решений. Решатель – это некоторая программа, помимо базового алгоритма использующая разнообразные эвристики, в том числе и эвристики удаления нерелевантных ограничений. Термин «прогнозная функция» обозначает семейство функций, параметрически зависящих от решателя *S*. Аргументами данных функций являются случайные выборки систем из множеств $\{\Phi^\alpha\}_{\alpha \in 2^{X^d}}$. Прогнозная функция является частично определенной на заданном множестве выборок и каждой «точке» своей области определения ставит в соответствие некоторое положительное рациональное число. Знание глобального минимума данной функции на ее области определения позволяет получить представление (прогноз) относительно минимального общего объема вычислений, требуемого для решения распараллеленной исходной задачи.

Случайные выборки из множеств $\{\Phi^\alpha\}_{\alpha \in 2^{X^d}}$ имеют смысл лишь в тех случаях, когда мощность данного множества (величина 2^d) слишком велика для его полного перебора. Поэтому при определении прогнозной функции имеет смысл рассматривать две ситуации: когда число 2^d больше, чем некоторая положительная константа ρ_0, и когда 2^d не превосходит ρ_0. За ρ_0 можно принять, например, число, сопоставимое с числом процессоров в кластере.

Далее полагаем, что все случайные выборки имеют фиксированный объем *q*. Каждому значению параметра $d \in \{0, 1, \ldots, n\}$, такому, что $2^d > \rho_0$, ставится в соответствие множество векторов $\{\alpha^1, \ldots, \alpha^q\}$, выбираемых из 2^{X^d} в соответствии с заданным на нем равномерным распределением, а также множество (выборка) систем $\Theta_d = \{\Phi^{\alpha_1}, ..., \Phi^{\alpha_q}\}$. Каждому значению параметра $d \in \{0, 1, \ldots, n\}$, такому, что $2^d \le \rho_0$, ставится в соответствие множество 2^{X^d} и множество систем $\{\Phi^\alpha\}_{\alpha \in 2^{X^d}}$.

Фиксируем некоторый решатель *S*. Обозначим через $t(\Phi')$ время работы (число битовых операций) решателя *S* на произвольном входе $\Phi' \in \Theta_d$. Введем в рассмотрение функцию

$$\tau_S(\Theta_d) = \sum_{\Phi' \in \Theta_d} t(\Phi').$$

Значением данной функции при каждом фиксированном $d \in \{0, 1, \ldots, n\}$ является суммарное время (число битовых операций) работы решателя *S* по всем системам из Θ_d.

Следует учитывать, что при некоторых значениях параметра d (например, при $d = 0$) системы Θ_d могут оказаться очень сложными для решателя S, и в этом случае время подсчета соответствующего значения прогнозной функции может превысить разумные границы. Для учета данного факта в рассмотрение вводится специальная функция $g(\Phi) = p(|\Phi|)$, где $p(\cdot)$ – некоторый полином, а через $|\Phi|$ обозначен объем двоичной кодировки исходной системы вида (4).

Допустим, что в соответствии с перечисленными правилами построено семейство выборок $\Theta = \{\Theta_d\}_{d\in\{0,1,\ldots,n\}}$ (при фиксированном ρ_0). Прогнозную функцию определим следующим образом:

$$T(\Theta_d) = \begin{cases} \dfrac{2^d}{q}\cdot\tau_S(\Theta_d),\ 2^d > \rho_0,\ \tau_S(\Theta_d) < g(|\Phi|), \\ \tau_S(\Theta_d),\ 2^d \le R_0,\ \tau_S(\Theta_d) < g(|\Phi|), \\ \infty,\ \tau_S(\Theta_d) \ge g(|\Phi|). \end{cases} \quad (6)$$

Запись «$T(\Theta_d) = \infty$» означает, что функция не определена на выборке Θ_d. Рациональное число $T(\Theta_d)$ является прогнозом общего объема битовых операций, требуемого для решения исходной SAT-задачи при декомпозиции исходной системы вида (4) на семейство систем $\{\Phi^\alpha\}_{\alpha\in 2^{X^d}}$. Тем самым задача прогнозного планирования оптимального по трудоемкости параллельного вычисления сводится к задаче минимизации функции T на множестве Dom $T \subseteq \Theta$. Знание глобального минимума функции T на Dom T дает представление о возможности параллельного решения SAT-задачи для КНФ C «за разумное время».

Далее приведен основной результат работы [15], который является непосредственным обобщением теоремы, доказанной в [2] в отношении SAT-задач.

Теорема 2 (см. [15]). Пусть Φ – произвольная булева система вида (4), $\Theta=\{\Theta_d\}_{d\in\{0,\,1,\,\ldots,\,n\}}$ – произвольное семейство выборок систем из множества $\{\Phi^\alpha\}_{\alpha\in 2^{X^d}}$. Для произвольной прогнозной функции T: $\Theta \to Q$ вида (6) справедливы следующие свойства:

1. Dom $T \neq \varnothing$.
2. Глобальный минимум T на Dom T находится в общем случае за время, ограниченное полиномом от $|\Phi|$.

В работе [15] было продемонстрировано, что технология крупноблочного параллелизма, использующая прогнозные функции, позволяет добиваться на задачах криптоанализа некоторых генераторов (в частности, порогового генератора) «сверхлинейного» ускорения. Как показано в [15], данный феномен является прямым следствием неполноты используемого решателя.

4. Пакет D-SAT

Для реализации описанной выше технологии крупноблочного распараллеливания применительно к SAT-задачам был разработан пакет прикладных программ Distributed-SAT (D-SAT) [16]. Пакет функционирует на вычислительном кластере под управлением инструментального комплекса DISCOMP [3, 17]. Описание данного программного комплекса здесь приводится.

Библиотека программ пакета D-SAT включает модуль расщепления, модуль SAT-решателя, модуль прогнозирования, модуль-анализатор и транспортный модуль. Модули пакета реализованы на языке C++, причем все они являются платформонезависимыми.

Основные входные данные модуля расщепления – это файл с исходной КНФ в формате DIMACS, диапазон значений параметра d (левая и правая границы) и фиксированное значение q, определяющее число КНФ в произвольной случайной выборке. Обозначим через d_* и d^* натуральные числа, определяющие соответственно верхнюю и нижнюю границы интервала, в котором изменяются значения d. Для каждого значения диапазона параметров декомпозиции строится отдельное семейство КНФ. В случае $q < 2^d$ из декомпозиционного семейства случайным образом выбираются q КНФ. Если $q = 0$ или $q \ge 2^d$, то выборкой является все рассматриваемое семейство. Результатом декомпозиции исходной КНФ является набор файлов с КНФ декомпозиционного семейства, представленными в формате DIMACS. Этот набор в процессе работы комплекса DISCOMP описывается в виде параллельного списка.

Вычислительное ядро модуля SAT-решателя составляет программа minisat [18], вообще говоря, могут использоваться различные версии данной программы. По входному файлу, в котором КНФ представлена в формате DIMACS, модуль SAT-решателя осуществляет поиск выполняющего данную КНФ набора либо выдает значение *NULL*, если таковой отсутствует. В выходной файл каждой копии модуля SAT-решателя записывается информация о результатах его работы. Если в процессе обработки параллельного списка для какой-либо КНФ найден выполняющий набор, то все вычисления прекращаются и запускается модуль-анализатор.

Модуль прогнозирования обрабатывает файлы с результатами работы модуля SAT-решателя и производит их статистический анализ, на основе которого строится прогноз оптимальных параметров декомпозиции. При этом дополнительно учитываются все транспортные расходы.

Аналитический модуль проверяет найденный выполняющий набор на корректность, вычисляет среднее время решения SAT-задач для списка КНФ, минимальное и максимальное время, а также сравнивает прогнозируемое время решения с реальным временем. Решение SAT-задачи и результаты анализа записываются в итоговый файл.

Процесс решения SAT-задачи состоит из двух этапов: а) прогнозирование наиболее оптимальных (с точки зрения вычислительных затрат) параметров декомпозиции; б) решение всех SAT-задач из декомпозиционного семейства, определяемого найденными на этапе прогнозирования параметрами декомпозиции.

На этапе прогнозирования выполняются следующие шаги (рис. 1):

1. На вход модулю расщепления подается файл с исходной КНФ в формате DIMACS, значения d_*, d^* и q. Модуль расщепления для каждого значения d строит соответствующую выборку КНФ с учетом значения q. Полученные файлы КНФ в формате DIMACS объединяются в параллельный список и передаются на модуль SAT-решателя.

2. Для каждой КНФ данного списка решается соответствующая SAT-задача – доказывается невыполнимость либо находится выполняющий набор.

3. После того как решены SAT-задачи для всех КНФ из списка, при помощи модуля прогнозирования производится статистический анализ полученных результатов и на его основе вычисляется d_0 – прогнозное значение оптимального параметра декомпозиции.

В процессе прогнозирования существует возможность того, что будет найден набор, выполняющий исходную КНФ. В этом случае решение задачи считается найденным и вычислительный процесс завершается.

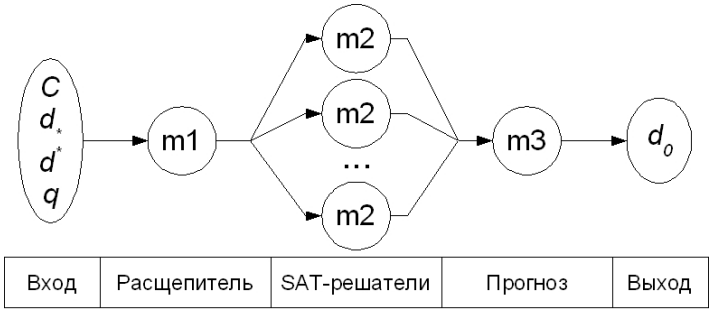

Рис. 1. Схема взаимодействия модулей пакета D-SAT на этапе прогнозирования

На этапе решения задачи выполняются следующие шаги (рис. 2):

1. На вход модулю расщепления подается файл с исходной КНФ в формате DIMACS, полагается $d_* = d^* = d_0$, где значение d_0 найдено на этапе прогнозирования, а также $q = 0$. Модуль расщепления строит семейство файлов КНФ C_1, …, C_k. Полученные КНФ объединяются в параллельный список и передаются на модуль SAT-решателя.

2. Для каждой КНФ полученного списка решается соответствующая SAT-задача – доказывается невыполнимость либо находится выполняющий набор.

3. Запускается модуль-анализатор.

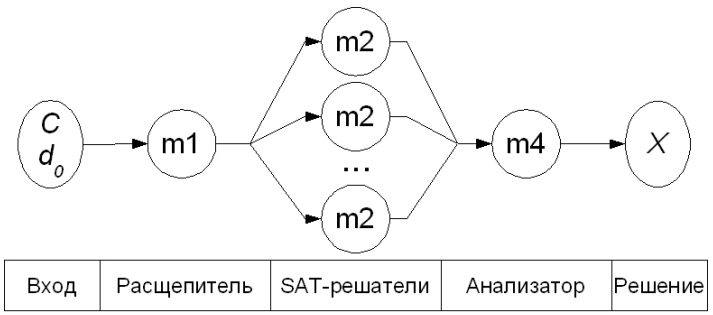

Рис. 2. Схема взаимодействия модулей пакета D-SAT на этапе решения

Обмен данными между вычислительными узлами кластера осуществляет специальный транспортный модуль пакета D-SAT, основной задачей которого является рассылка КНФ из параллельного списка. Каждая КНФ представляется в виде файла в формате DIMACS, состоящего из трех частей:

1) заголовок вида «*p cnf* [*число переменных*] [*число дизъюнктов*]»;

2) список из d однолитеральных дизъюнктов, построенных по соответствующим значениям переменных декомпозиции;

3) список дизъюнктов исходной КНФ.

В таком представлении у всех КНФ декомпозиционного семейства первая и третья части одинаковы, а различаются лишь вторые части. Назовем файлом различий файл со списком однолитеральных дизъюнктов произвольной КНФ семейства.

Работа транспортного модуля включает этап препроцессинга, в ходе которого на каждый вычислительный узел передается файл с исходной КНФ. В дальнейшем на вычислительные узлы передаются только файлы различий КНФ. Каждая отдельная SAT-задача формируется на основе информации из файла различия и файла с исходной КНФ. Описанная схема позволяет значительно сократить транспортные расходы.

5. Формирование декомпозиционных множеств; оптимизация параллельных планов в задачах логического криптоанализа; прогноз параллельного логического криптоанализа генератора A5/1

Довольно очевидный факт состоит в том, что значительное влияние на эффективность параллельного решения SAT-задачи может оказывать выбор декомпозиционного множества X^d. Здесь мы приводим некоторые общие схемы построения декомпозиционных множеств, дающих оптимальные по вычислительным затратам параллельные планы[1]. В качестве иллюстрации описываемого подхода рассматривается задача логического криптоанализа широко известной системы потокового шифрования A5/1 (рис. 3).

Далее мы рассматриваем задачу криптоанализа только генератора ключевого потока в данном шифре, оставляя за скобками собственно протокол. Описание генератора взято из [19]. В соответствии с этим описанием, в генераторе A5/1 используются 3 регистра сдвига с линейной обратной связью (РСЛОС, LFSR), задаваемые следующими полиномами обратной связи: РСЛОС 1: $X^{18} + X^{17} + X^{16} + X^{13} + 1$ (длина регистра 19 бит); РСЛОС 2: $X^{21} + X^{20} + 1$ (длина регистра 22 бита); РСЛОС 3: $X^{22} + X^{21} + X^{20} + X^7 + 1$ (длина регистра 23 бита).

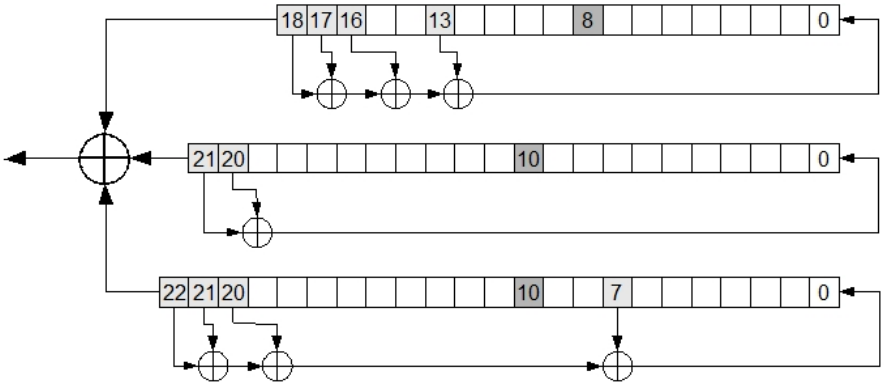

Рис. 3. Схема работы генератора ключевого потока шифра A5/1

В каждом такте могут сдвигаться не все регистры. Сдвиг регистра с номером m, $m \in \{1, 2, 3\}$, происходит, если значение функции $\chi_m\left(b_s^1, b_s^2, b_s^3\right)$ равно 1, и не происходит, если значение данной функции равно 0. Через b_s^1, b_s^2, b_s^3 здесь обозначены значения т.н. «серединных битов» текущего шага, то есть битов, находящихся в данный момент в девятой ячейке первого РСЛОС, и в ячейках с номером «11» РСЛОС2 и РСЛОС3 (данные ячейки на рис. 3 отмечены более темной заливкой, нумерация ячеек ведется справа налево). Функция $\chi_m(\cdot)$ определяется следующим образом:

$$\chi_m\left(b_s^1, b_s^2, b_s^3\right) = \begin{cases} 1, b_s^m = \text{majority}\left(b_s^1, b_s^2, b_s^3\right), \\ 0, b_s^m \neq \text{majority}\left(b_s^1, b_s^2, b_s^3\right), \end{cases}$$

где $\text{majority}(x, y, z) = x \cdot y \vee x \cdot z \vee y \cdot z$ (функция большинства).

Проблема построения оптимальной декомпозиции для параллельного решения задачи криптоанализа A5/1 оказалась весьма нетривиальной. Были исследованы различные схемы построения декомпозиционных множеств. Первые стратегии основывались на идеях, ранее успешно примененных в параллельном криптоа-

[1] Термин «оптимальность» здесь, конечно же, понимается в отношении конкретного SAT-решателя.

нализе ряда генераторов (пороговый, суммирующий, Гиффорда [3]). В частности, использовалась схема подстановки подряд идущих битов (в естественной нумерации) инициализирующей последовательности. Тем самым, например, при $d = 16$ в декомпозиционное множество включались переменные, кодирующие начальные состояния первых 16 ячеек РСЛОС1. При этом, исходя из некоторых «разумных» предположений, сначала строилось базовое декомпозиционное множество, а затем предпринимались попытки его усечения на основе значений соответствующих прогнозных функций. Ниже представлена схема (рис. 4) построения базового декомпозиционного множества X^d, состоящего из 31 переменной.

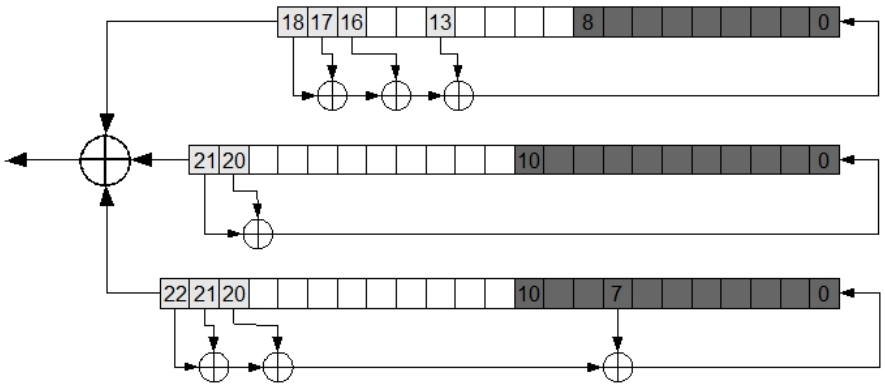

Рис. 4. Схема построения декомпозиционного множества из 31 переменной (X^d)

В соответствии с данной схемой предлагается включить в X^d переменные, кодирующие начальные состояния ячеек регистров, начиная с первых ячеек до ячеек, содержащих серединные биты (соответствующие ячейки на рис. 4 отображены темной заливкой). Иными словами, имеем декомпозиционное множество

$$X^d = \{x_1, \ldots, x_9, x_{20}, \ldots, x_{30}, x_{42}, \ldots, x_{52}\}. \quad (7)$$

Данный выбор продиктован следующими естественными соображениями. Произвольная фиксация битов из построенного таким образом множества X^d индуцирует точные значения серединных битов для значительного числа последующих состояний всех трех регистров. Но серединные биты являются наиболее информативными битами, поскольку именно они определяют, какая ветвь соответствующего условия (значение функции большинства) имеет место.

Довольно неожиданным оказался тот факт, что построенное в соответствии с (7) множество X^d не удалось существенно уменьшить за счет технологии прогнозных функций. В проводимых статистических экспериментах для каждого варианта декомпозиционного множества и начального фрагмента ключевого потока некоторой фиксированной длины строились случайные выборки объемом 1000 КНФ. Для каждой такой выборки считалось значение прогнозной функции – среднее значение времени решения SAT-задач по выборке, умноженное на мощность пространства перебора (см. п. 3). Во всех вычислительных экспериментах SAT-задачи решались SAT-решателем minisat версии 1.2. Каждая задача решалась на одном ядре процессора Intel E8400.

Далее в табл. 1 приведены значения среднего времени решения SAT-задач для спектра различных длин выхода и для различных вариантов формирования декомпозиционных множеств. Все эти варианты тем не менее в концептуальном смысле были близки к (7). Например, декомпозиционное множество, состоящее из 28 переменных, формировалось в соответствии со схемой на рис. 5 (в контексте оговоренных выше обозначений).

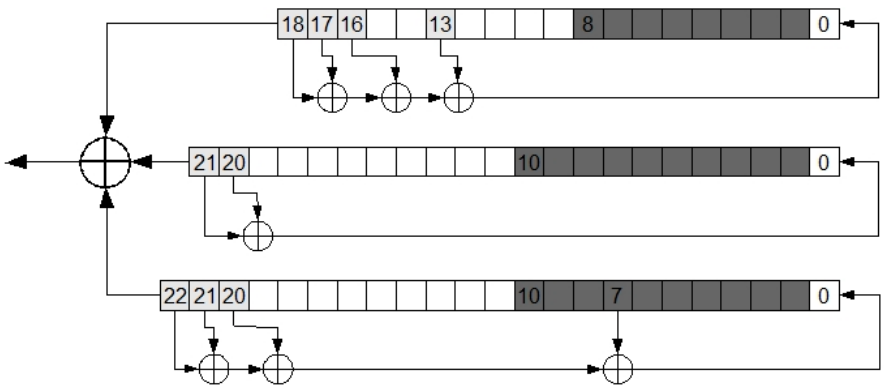

Рис. 5. Схема построения декомпозиционного множества из 28 переменных

Наилучшие итоговые результаты дало декомпозиционное множество, обозначаемое далее через X_*^d:

$$X_*^d = \{x_1,...,x_9,x_{21},...,x_{30},x_{42},...,x_{52}\}. \quad (8)$$

Таким образом, данное множество формировалось по следующей схеме (рис. 6):

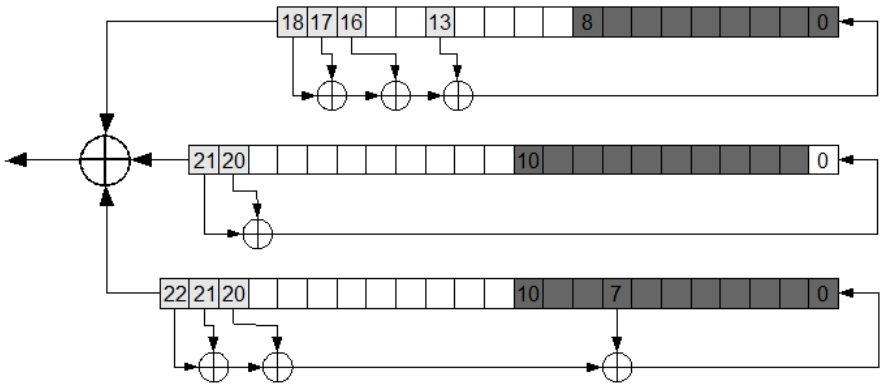

Рис. 6. Схема построения декомпозиционного множества из 30 переменных (X_*^d)

Таблица 1

Среднее время решения SAT-задач на одном ядре процессора Intel E8400 при различных вариантах декомпозиции и разных длинах выхода (объем каждой случайной выборки – 1000 КНФ, решатель minisat 1.2)

Число переменных	Длина выхода				
	144	160	176	192	208
28	4,241943	4,03173	4,79874	5,492782	5,1268
29	2,041447	2,062648	2,068609	1,888878	2,139253
30	0,848397	0,849213	0,948331	0,870502	1,006667
31	0,423078	0,428842	0,471429	0,420562	0,515448
32	0,229554	0,252899	0,320812	0,253239	0,267752
33	0,147385	0,151029	0,161326	0,156497	0,171918

Прогноз эффективности решения параллельной задачи криптоанализа требует также учета времени, затрачиваемого на пересылку файлов к процессорам/ядрам кластера (транспортные расходы). Благодаря схеме передачи данных, используемой в D-SAT (см. п. 4), эти затраты крайне незначительны – напомним, что «формулировкой» каждой новой задачи, передаваемой от центрального процессора на произвольный процессор/ядро, является двоичный вектор, длина которого равна мощности декомпозиционного множества. В табл. 2 приведены суммарные транспортные расходы на пересылку векторов длин 28 – 33 для двух различных типов соединений между вычислительными узлами (Gigabit Ethernet и Infiniband), наиболее часто используемых в современных кластерах. Данный прогноз был осуществлен на кластере Blackford ИДСТУ СО РАН [20] относительно системы Gigabit Ethernet и экстраполирован на систему Infiniband.

Таблица 2

Транспортные расходы при различных типах соединений между вычислительными узлами

Количество пересылаемых файлов	Транспортные расходы с интерконнектом Gigabit Ethernet, ч	Транспортные расходы с интерконнектом Infiniband, ч
2^{28}	1,5	0,75
2^{29}	3	1.5
2^{30}	6	3
2^{31}	12	6
2^{32}	24	12
2^{33}	48	24

Как уже отмечалось выше, все численные эксперименты проводились на процессоре Intel E8400. В табл. 3 приведены сравнительные характеристики данного процессора и процессора Intel E5472.

Кластер Т-60 [21] на момент написания данной статьи занимает 1-е место в списке 50-ти самых мощных суперкомпьютеров в СНГ [22]. Т-60 состоит из 1250 четырехъядерных процессоров E5472.

Таблица 3

Характеристики процессоров

Название процессора	Intel E8400	Intel E5472
Число ядер	2	4
Частота ядра	3,0 ГГц	3,0 ГГц
Частота шины	1333 МГц	1600 МГц
Технология производства	45 нм	45 нм
Кэш L2	6 Мб	12 Мб

Из данной таблицы видно, что мощность одного ядра процессора Intel E8400 сопоставима с мощностью одного ядра процессора Intel E5472 (незначительное отличие лишь в частоте шины). Исходя из этого факта, мы сочли возможным напрямую экстраполировать все приведенные выше результаты численных экспериментов применительно к процессу решения задачи логического криптоанализа генератора A5/1 на кластере Т-60. Наилучшие результаты приведены ниже.

1. Для декомпозиционного множества X^d_*, построенного в соответствии с (8), и 144 бит выходной последовательности прогноз времени решения задачи криптоанализа A5/1 (нахождение инициализирующей последовательности) на Т-60: 54 часа при интерконнекте Infiniband, 57 часов при интерконнекте Gigabit Ethernet.

2. Для декомпозиционного множества X^d, построенного в соответствии с (7), и 192 бит выходной последовательности прогноз времени решения задачи криптоанализа A5/1 на Т-60: 56 часов при интерконнекте Infiniband, 62 часа при интерконнекте Gigabit Ethernet.

Заключение

Описанные в данной работе подходы к обращению дискретных функций на многопроцессорных системах представляются весьма перспективными ввиду впечатляющего роста производительности вычислительной техники. Основной причиной этого, конечно же, следует считать многоядерные технологии, динамичное развитие которых делает возможным построение кластеров с гигантскими вычислительными мощностями (Т-60 [21], СКИФ Cyberia, [23]). Рассматриваемые в настоящей работе задачи и подходы к их решению интересны своей высокой масштабируемостью (прямая зависимость эффективности решения от числа процессоров/ядер).

В качестве дальнейших перспектив данного направления следует отметить удачно вписывающуюся в его контекст проблему анализа криптографических хэш-функций. Позитивной особенностью SAT-задач, кодирующих проблемы обращения хэш-функций, является тот факт, что соответствующие КНФ выполнимы более чем на одном наборе (как правило, число выполняющих наборов есть некоторая экспонента от длины входа). Это делает возможным использование новых алгоритмических подходов для решения такого рода SAT-задач (например, алгоритмы, базирующиеся на идеологии локального поиска).

ЛИТЕРАТУРА

1. *Цейтин Г.С.* О сложности вывода в исчислении высказываний // Зап. научн. семинаров ЛОМИ АН СССР. 1968. Т. 8. С. 234 – 259.
2. *Заикин О.С.*, *Семенов А.А.* Технология крупноблочного параллелизма в SAT-задачах // Проблемы управления. 2008. № 1. С. 43 – 50.
3. *Заикин О.С.*, *Семенов А.А.*, *Сидоров И.А.*, *Феоктистов А.Г.* Параллельная технология решения SAT-задач с применением пакета прикладных программ D-SAT // Вестник ТГУ. Приложение. 2007. № 23. С. 83 – 95.
4. *Robinson J.A.* A machine-oriented logic based on the resolution principle // J. ACM. 1965. V. 12. No. 1. P. 23 – 41. [Перевод: *Робинсон Дж.А*. Машинно-ориентированная логика, основанная на принципе резолюций // Кибернетический сборник. Новая серия. М.: Мир, 1970. Вып. 7. С. 194 – 218.]
5. *Haken A.* The intractability of resolution // Theoretical Computer Science. 1985. V. 39. P. 297 – 308. [Перевод: *Хакен А.* Труднорешаемость резолюций // Кибернетический сборник. Новая серия. М.: Мир, 1991. Вып. 28. С. 179 – 194.]
6. *Tseitin G.* On the complexity of derivation in propositional calculus // In Studies in Constructive Mathematics and Mathematical Logic, part 2. 1968. P. 115 – 125. Reprinted in J. Siekmann and G. Wrightson (editors). Automation of reasoning. Berlin: Springer Verlag, 1983. V. 2. P. 466 – 483.
7. *Семенов А.А.*, *Буранов Е.В.* Погружение задачи криптоанализа симметричных шифров в пропозициональную логику // Вычислительные технологии (совместный выпуск с журналом «Региональный вестник Востока»). 2003. Т. 8. С. 118 – 126.

8. *Семенов А.А.* О сложности обращения дискретных функций из одного класса // Дискретный анализ и исследование операций. 2004. Т. 11. № 4. С. 44 – 55.
9. *Simon J.* On Some Central Problems in Computational Complexity: Doctoral Thesis, Dept. of Computer Science, Cornel University, Ithaca, NY, 1975.
10. *Гэри М.*, *Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
11. *Семенов А.А.* Логико-эвристический подход в криптоанализе генераторов двоичных последовательностей // Труды Междунар. науч. конф. ПАВТ'07. Челябинск: ЮУрГУ, 2007. Т. 1. С. 170 – 180.
12. *Cook S.A.* The complexity of theorem-proving procedures // Proc. 3rd Ann. ACM Symp. on Theory of Computing, Association for Computing Machinery, Ohio, 1971. P. 151 – 159. [Перевод: *Кук С.А.* Сложность процедур вывода теорем // Кибернетический сборник. Новая серия. М.: Мир, 1975. Вып. 12. С. 5 – 15.]
13. *Буранов Е.В.* Программная трансляция процедур логического криптоанализа симметричных шифров // Вестник ТГУ. Приложение. 2004. № 9 (1). С. 60 – 65.
14. *Буров П.С.*, *Игнатьев А.С.*, *Отпущенников И.В.* Программная трансляция алгоритмов в логические выражения в задачах диагностирования дискретных систем // Материалы IX школы-семинара «Математическое моделирование и информационные технологии». Иркутск, 2007. С. 33 – 35.
15. *Семенов А.А.*, *Заикин О.С.* Неполные алгоритмы в крупноблочном параллелизме комбинаторных задач // Труды Междунар. науч. конф. ПАВТ'08. СПб., 2008. С. 232 – 244.
16. *Заикин О.С.* Пакет прикладных программ Distributed-SAT. Свидетельство об официальной регистрации программы ЭВМ № 2008610423, опубл. 23.01.2008.
17. *Феоктистов А.Г.*, *Сидоров И.А.* Языковые средства описания распределенных вычислений в инструментальном комплексе DISCOMP // Труды Междунар. науч. конф. ПАВТ'08. СПб., 2008. С. 488 – 493.
18. MiniSat [http://minisat.se/MiniSat.html]
19. *Biryukov A.*, *Shamir A.*, *Wagner D.* Real Time Cryptanalysis of A5/1 on a PC. Fast Software Encryption Workshop 2000. April 10 – 12, 2000. New-York City.
20. *Суперкомпьютерный* центр ИДСТУ СО РАН [http://www.mvs.icc.ru]
21. *Научно-исследовательский* вычислительный центр МГУ им. М.В. Ломоносова [http://parallel.ru/cluster/]
22. *Список* суперкомпьютеров топ-50 [http://supercomputers.ru/]
23. *Межрегиональный* вычислительный центр ТГУ [http://skif.tsu.ru]

ПРИКЛАДНАЯ ТЕОРИЯ КОДИРОВАНИЯ

DOI 10.17223/20710410/2/26

УДК 519.72

АРИФМЕТИЧЕСКОЕ КОДИРОВАНИЕ СООБЩЕНИЙ С ИСПОЛЬЗОВАНИЕМ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ[1]

В.Н. Потапов

Институт математики им. С.Л. Соболева СО РАН, г. Новосибирск

E-mail: vpotapov@math.nsc.ru

Предлагается модификация метода арифметического кодирования сообщений, использующая некоторую случайную последовательность как секретный ключ. Доказано, что предлагаемый метод достигает теоретической границы сжатия текстов не только при кодировании обыкновенных сообщений, но и при кодировании частично определённых данных.

Ключевые слова: *арифметическое кодирование*, *сжатие данных*, *энтропия источника сообщений*, *частично определённые данные.*

Арифметическое кодирование является одним из наиболее известных методов, применяемых для сжатия текстов. В качестве математической модели текста обычно рассматривают множество конечных слов в некотором алфавите A с заданным на множестве слов распределением вероятностей. Кодированием называется инъективное отображение f множества слов в алфавите A в множество двоичных слов. Основными критериями эффективности метода кодирования являются степень сжатия текста и трудоёмкость кодирования и декодирования.

Идея арифметического кодирования, которая будет изложена ниже, была высказана П. Элайесом. Й. Риссанен [1, 2] предложил первый эффективный алгоритм вычисления арифметического кода с почти линейной относительно длины слова трудоёмкостью кодирования и декодирования. В дальнейшем этот метод был развит в многочисленных работах, в частности в [3, 4]. Подробное описание эффективного алгоритма арифметического кодирования имеется в [5].

В настоящей работе предлагается новое воплощение идеи арифметического кодирования с использованием в качестве параметра произвольной случайной последовательности. Новый метод обеспечивает такую же теоретическую степень сжатия сообщений, как и классическая реализация, при этом позволяя использовать произвольную случайную последовательность чисел из интервала (0,1) как секретный ключ. Кроме того, предлагаемый метод позволяет эффективно сжимать не только обычные, но и частично определённые данные, введённые в работах Л.А. Шоломова [6, 7].

Источником сообщений называется пара из конечного алфавита A и распределения вероятностей P на множестве A^* конечных слов в алфавите A. Распределение вероятностей должно удовлетворять естественным соотношениям $\sum P(wa)=P(w)$, где сумма берётся по всем продолжениям wa, $a\in A$, слова $w\in A^*$, и $\sum P(w) = 1$, где сумма берётся по всем словам одинаковой длины. Источник сообщений называется источником без памяти (источником Бернулли), если вероятность буквы в слове не зависит от контекста, т. е. $P(a_1, \ldots, a_n) = P(a_1)\ldots P(a_n)$. Величина $I(w)= -\log P(w)$ называется сложностью слова $w\in A^*$ относительно источника сообщений (A, P). Из теоремы Шеннона (см., например, [8]) следует, что величина $I(w)$ является длиной двоичного кода слова при оптимальном префиксном кодировании.

Арифметическое кодирование сообщений заключается в следующем. Все слова $w\in A^n$ упорядочим лексикографически и для каждого слова определим величины $L(w)= \sum P(x)$, где сумма берётся по всем словам $x\in A^n$, лексикографически меньшим, чем слово w, и $R(w) = L(w) + P(w)$. Разделим полуинтервал [0,1) на полуинтервалы $i(w) = [L(w), R(w))$. В каждом полуинтервале $i(w)$ длины $P(w)$ найдётся двоично-рациональное число $q(w)$ со знаменателем, не большим, чем $2^{\lceil -\log P(w)\rceil}$, поскольку разность между ближайшими числами с таким знаменателем не превосходит длины полуинтервала. В качестве кода $f(w)$ слова $w\in A^n$ рассмотрим

[1] Исследование выполнено при финансовой поддержке РФФИ (проект 08-01-00671).

двоичную запись числителя дроби $q(w)$ длиной $\lceil -\log P(w) \rceil$, в случае необходимости перед двоичной записью числителя следует приписать недостающие нули. Полученное отображение f является инъективным на множестве A^n по построению, причём для длины кода справедливо неравенство $|f(w)| \le I(w) + 1$, т. е. длина кода каждого слова является оптимальной с точностью до неизбежного округления к ближайшему целому. Нетрудно видеть, что отображение f будет инъективным и на A^*, если каждое продолжение произвольного слова w имеет вероятность меньше, чем $P(w)/2$. В противном случае отображение f можно преобразовать в инъективное и даже префиксное кодирование без существенного увеличения длины кода, приписывая $Cod(n)$ в качестве префикса к $f(w)$ при $w \in A^n$, где Cod – префиксное кодирование натуральных чисел, удовлетворяющее свойству $|Cod(n)| = \log n(1 + o(1))$. Примеры таких префиксных кодов натурального ряда имеются, например, в [8, 9].

Предлагается следующая модификация арифметического кодирования. Пусть $x_1, \ldots, x_n, \ldots$ – последовательность чисел из интервала (0,1). Рассмотрим слово $w \in A^n$. Пусть m – номер первого из чисел последовательности, попавших в полуинтервал $i(w)$, т.е. $x_m \in i(w)$. В качестве кода слова w рассмотрим $g_x(w) = Cod(m)$. Нетрудно видеть, что отображение g_x является инъективным отображением на A^n.

Теорема 1. Пусть $\xi_1, \ldots, \xi_m, \ldots$ – последовательность независимых равномерно распределённых на (0,1) случайных величин. Тогда $\mathbf{E}|g_\xi(w)| \le I(w)(1 + o(1))$ при $|w| \to \infty$.

Доказательство. Пусть $A \subseteq (0, 1)$. По условию $P(\xi_i \in A) = P(A)$ – мера Лебега множества A для всех $i \in 1, \ldots, n$ и $P(\xi_1, \ldots, \xi_{n-1} \notin A, \xi_n \in A) = (1 - P(A))^{n-1} P(A)$. Рассмотрим случайную величину m на измеримых подмножествах $A \subseteq (0, 1)$, равную номеру первой из случайных ξ_i, такой, что $\xi_i \in A$. Очевидно $\mathbf{E}(m) = \sum_{n=1}^{\infty} n(1 - P(A))^{n-1} P(A) = 1/P(A)$. Поскольку $\log t$ – выпуклая вверх функция, имеем $\mathbf{E}(\log m) \le \log(\mathbf{E}(m))$. Пусть $A = i(w)$. Тогда $\mathbf{E}|g_\xi(w)| = \mathbf{E}|Cod(m)| = (1 + o(1))\mathbf{E}(\log m) \le (1 + o(1)) \log(\mathbf{E}(m)) = I(w)(1 + o(1))$ при $|w| \to \infty$. Теорема доказана.

Так же как известное арифметическое кодирование f, предлагаемое кодирование g_ξ можно преобразовать в префиксное без асимптотического увеличения длины кода. Из теоремы Шеннона следует, что полученная в теореме 1 оценка не улучшаема ни для какого префиксного кодирования, так как средняя длина префиксного кода слова не может быть меньше величины $I(w)$.

Описанная выше модификация арифметического кодирования может быть использована для кодирования частично определённых данных. Задача кодирования частично определённых данных состоит в следующем. Пусть $B = 2^A$ – множество подмножеств алфавита A. Слово $w = a_1 \ldots a_n$ называется доопределением сообщения $W = b_1 \ldots b_n$, если $a_i \in b_i$ для всех $i = 1 \ldots n$. Кодированием частично определённых данных называется отображение F из множества B^* в множество двоичных слов, обладающее свойством: найдётся такая функция (декодирование) F^o, действующая из множества двоичных слов в множество слов в алфавите A, что $F^o(F(W))$ является доопределением слова $W \in B^*$. Таким образом, результатом декодирования будет не исходное частично определённое сообщение, а некоторое его доопределение. Источник частично определённых сообщений без памяти определяется как пара (B, P), где P – некоторое распределение вероятностей на множестве B.

В работе [7] доказан аналог теоремы Шеннона для источников (без памяти) частично определённых сообщений. А именно, доказано, что величина $H = \min\left(-\sum_{b \in B} P(b) \log\left(\sum_{a \in b} Q(a)\right)\right)$, где минимум берётся по всевозможным распределениям вероятностей Q на алфавите A, является нижней гранью стоимости кодирования, т. е. средней длины кода в расчёте на букву исходного недоопределённого сообщения $W \in B^*$. Естественно определить сложность $b \in B$ относительно (B, P) как $I(b) = -\log \sum_{a \in b} Q(a)$, а сложность $I(W)$ сообщения $W = b_1 \ldots b_n$ – как сумму сложностей $I(b_i)$.

Пусть Q – распределение вероятностей, на котором достигается минимум H. Для каждого слова $w \in A^n$ определим полуинтервал $i(w)$ в соответствии с распределением Q. Каждому сообщению $W \in B^*$ поставим в соответствие множество $i(W) = \cup\, i(w)$, где объединение берётся по всем словам $w \in A^*$, доопределяющим сообщение $W \in B^*$. Пусть $x_1, \ldots, x_n, \ldots$ – последовательность чисел из интервала (0,1). Пусть m – номер первого из чисел последовательности, попавших в множество $i(W)$, т.е. $x_m \in i(W)$. В качестве кода сообщения W рассмотрим $G_x(W) = Cod(m)$. Нетрудно видеть, что отображение G_x является инъективным отображением на B^n.

Теорема 2. Пусть $\xi_1, \ldots, \xi_m, \ldots$ – последовательность независимых равномерно распределённых на (0,1) случайных величин. Тогда $\mathbf{E}|G_\xi(W)| \le I(W)(1 + o(1))$ при $|W| \to \infty$.

Доказательство теоремы 2 аналогично доказательству теоремы 1. Из [7] следует, что предложенное кодирование обеспечивает асимптотически минимальную длину кодовых слов.

ЛИТЕРАТУРА

1. *Rissanen J.* Generalized Kraft inequality and arithmetic coding // IBM J. Res. Develop. 1976. V. 20. No. 3. P. 198 – 203.
2. *Rissanen J.*, *Langdon G.* Universal modelling and coding // IEEE Trans. Inform. Theory. 1981. V. IT-27. No. 1. P. 12 – 23.
3. *Рябко Б.Я.*, *Фионов А.Н.* Эффективный метод арифметического кодирования для источников с большими алфавитами // Проблемы передачи информации. 1999. Т. 35. Вып. 4. С. 95 – 108.
4. *Witten I.H.*, *Neal R.M.*, *Cleary J.G.* Arithmetic coding for data compression // Commun. ACM. 1987. V. 30. No. 6. P. 520 – 540.
5. *Потапов В.Н.* Арифметическое кодирование вероятностных источников // Дискретная математика и её приложения: Сб. лекций молодёжных научных школ по дискретной математике и её приложениям II. М.: Изд-во центра прикладных исследований при ММФ МГУ, 2001. С. 59 – 70.
6. *Шоломов Л.А.* О мере информации нечётких и частично определённых данных // Докл. РАН. 2006. Т. 140. № 1. С. 321 – 325.
7. *Шоломов Л.А.* Сжатие частично определённой информации // Нелинейная динамика и управление. Вып. 4. М.: Физматлит, 2004. С. 385 – 399.
8. *Потапов В.Н.* Теория информации. Кодирование дискретных вероятностных источников. Новосибирск: Изд. центр НГУ, 1999.
9. *Левенштейн В.И.* Об избыточности и замедлении разделимого кодирования натуральных чисел // Проблемы кибернетики. М.: Наука, 1968. Вып. 20. С. 173 – 179.

СВЕДЕНИЯ ОБ АВТОРАХ

БЕСПАЛОВ Дмитрий Викторович, научный сотрудник Института динамики систем и теории управления СО РАН, г. Иркутск. E-mail: **bespalov@altrixsoft.com**

БУРОВ Павел Сергеевич, аспирант Института динамики систем и теории управления СО РАН, г. Иркутск. E-mail: **evle.zzz@gmail.com**

БЫЛКОВ Даниил Николаевич, аспирант Московского государственного института радиотехники, электроники и автоматики. E-mail: **Bilkov@gmail.com**

ГЛУХОВ Михаил Михайлович, доктор физико-математических наук, академик Академии криптографии РФ, г. Москва. E-mail: **glukhovmm@rambler.ru**

ГРИШИН Анатолий Михайлович, старший научный сотрудник, кандидат технических наук, доцент Института криптографии, связи и информатики Академии ФСБ России, г. Москва. E-mail: **av123470@comtv.ru**

ДИМИТРИЕВ Юрий Константинович, профессор, доктор технических наук, ведущий научный сотрудник Института физики полупроводников СО РАН, г. Новосибирск. E-mail: **dimi@isp.nsc.ru**

ДУЛЬКЕЙТ Владимир Игоревич, аспирант Омского государственного университета.

ЕГОРУШКИН Олег Игоревич, аспирант кафедры математического моделирования и информатики Красноярского государственного аграрного университета.

ЗАИКИН Олег Сергеевич, аспирант Института динамики систем и теории управления СО РАН, г. Иркутск. E-mail: **oleg.zaikin@icc.ru**

ЗОЛОТАРЕВ Вячеслав Владимирович, кандидат технических наук, доцент кафедры безопасности информационных технологий Сибирского государственного аэрокосмического университета им. акад. М.Ф. Решетнева, г. Красноярск. E-mail: **amida@land.ru**

ИВАЩЕНКО Евгения Александровна, аспирант Донецкого национального университета. E-mail: **eugenia 1000@rambler.ru**

КАЛУГИН-БАЛАШОВ Дмитрий Андреевич, студент 5 курса Института космических и информационных технологий Сибирского федерального университета, г. Красноярск.

КАЧАНОВ Марк Александрович, студент Томского государственного университета.

КОВАЛЕВ Александр Михайлович, профессор, доктор физико-математических наук, директор Института прикладной математики и механики НАН Украины, г. Донецк, чл.-корр. НАНУ. E-mail: **kovalev@iamm.ac.donetsk.ua**

КОЗЛОВСКИЙ Валерий Анатольевич, старший научный сотрудник, кандидат физико-математических наук, заведующий лабораторией дискретной математики и прикладной алгебры Института прикладной математики и механики НАН Украины, г. Донецк. E-mail: **kozlovskii@iamm.ac.donetsk.ua**

КОЛЕГОВ Денис Николаевич, аспирант Томского государственного университета. E-mail: **kden@sibmail.com**

КУКАРЦЕВ Анатолий Михайлович, руководитель Ресурсного центра лицея № 3, аспирант Сибирского государственного аэрокосмического университета им. акад. М.Ф. Решетнева, г. Красноярск. E-mail: **amkukarcev@mail.ru**

КЫРОВ Владимир Александрович, доцент Горно-Алтайского государственного университета. E-mail: **kfizika@gasu.ru**

ЛАПШИН Владимир Владимирович, аспирант Томского государственного университета. E-mail: **alter@mail.tsu.ru**

МЕЛЕНТЬЕВ Виктор Александрович, кандидат технических наук, старший научный сотрудник Института физики полупроводников СО РАН, г. Новосибирск. E-mail: **melva@isp.nsc.ru**

МУРАДОВ Роман Мохуббатович, аспирант Горно-Алтайского государственного университета. E-mail: **kfizika@gasu.ru**

ПАРВАТОВ Николай Георгиевич, кандидат физико-математических наук, доцент кафедры защиты информации и криптографии Томского государственного университета. E-mail: **parvatov@mail.tsu.ru**

ПАУТОВ Павел Александрович, аспирант Томского государственного университета. E-mail: **__Pavel__ @mail.ru**

ПЕСТУНОВА Тамара Михайловна, кандидат технических наук, проректор по информатизации Новосибирского государственного университета экономики и управления. E-mail: ptm@nsaem.ru

ПОЗДЕЕВ Анатолий Геннадьевич, аспирант Томского государственного университета. E-mail: **anatoliy. pozdeev@vitasw.com**

ПОПОВ Алексей Михайлович, доктор физико-математических наук, заведующий кафедрой высшей математики факультета информатики и систем управления Сибирского государственного аэрокосмического университета им. акад. М.Ф. Решетнева, г. Красноярск. E-mail: **alexey_m_popov@newmail.ru**

ПОТАПОВ Владимир Николаевич, кандидат физико-математических наук, старший научный сотрудник Института математики им. С.Л. Соболева СО РАН, г. Новосибирск. E-mail: **vpotapov@math.nsc.ru**

ПУДОВКИНА Марина Александровна, кандидат физико-математических наук, доцент кафедры криптологии и дискретной математики Московского инженерно-физического института. E-mail: **maricap@ rambler.ru**

РОДИОНОВА Зинаида Валерьевна, аспирант Новосибирского государственного университета экономики и управления. E-mail: **rodionova@nsaem.ru**

САФОНОВ Константин Владимирович, доктор физико-математических наук, профессор кафедры прикладной математики Института космических и информационных технологий Сибирского федерального университета, г. Красноярск. E-mail: **safonovkv@rambler.ru**

СЕМЕНОВ Александр Анатольевич, кандидат технических наук, ведущий научный сотрудник Института динамики систем и теории управления СО РАН, г. Иркутск. E-mail: **biclop@rambler.ru**

СКОБЕЛЕВ Владимир Геннадиевич, профессор, доктор технических наук, ведущий научный сотрудник Института прикладной математики и механики НАН Украины, г. Донецк. E-mail: **skbv@iamm.ac.donetsk.ua**

ТУЖИЛИН Михаил Эльевич, старший научный сотрудник, кандидат физико-математических наук, доцент кафедры фундаментальной и прикладной математики Российского государственного гуманитарного университета, г. Москва. E-mail: **mtmt@rambler.ru**

ФАЙЗУЛЛИН Рашит Тагирович, доктор технических наук, профессор Омского государственного университета. E-mail: **r.t.faizullin@mail.ru**

ФОМИЧЁВ Владимир Михайлович, старший научный сотрудник, кандидат физико-математических наук, доцент кафедры № 42 Московского инженерно-физического института. E-mail: fomichev@nm.ru

ХМЕЛЬНОВ Алексей Евгеньевич, кандидат технических наук, заведующий отделом Института динамики систем и теории управления СО РАН, г. Иркутск. E-mail: **alex@icc.ru**

ХНЫКИН Иван Геннадьевич, аспирант Омского государственного университета.

ЧЕРНУШЕНКО Юрий Николаевич, студент Томского государственного университета.

ШЕСТАКОВ Василий Сергеевич, магистрант Сибирского государственного аэрокосмического университета им. акад. М.Ф. Решетнева, г. Красноярск. E-mail: **shockk@mail.ru**

ШИРКОВА Елена Андреевна, инженер группы контроля, анализа и информационной безопасности отдела 701 ОАО «Информационные спутниковые системы», аспирант кафедры безопасности информационных технологий Сибирского государственного аэрокосмического университета им. акад. М.Ф. Решетнева, г. Красноярск.

ЩЕРБАК Владимир Федорович, кандидат физико-математических наук, старший научный сотрудник Института прикладной математики и механики НАН Украины, г. Донецк. E-mail: **shvf@iamm.ac.donetsk.ua**

АННОТАЦИИ СТАТЕЙ НА АНГЛИЙСКОМ ЯЗЫКЕ

Bilkov D.N. **THE UNICITY DISTANCE OF A COORDINATE SEQUENCES FAMILY GENERATED BY COMPLICATIONS OF LRS OVER A GALOIS RING.** The existence of equivalent sequences for complicated coordinate LRS over a Galois ring is discussed in this paper. The minimal length at which the output sequences are distinguishable is also investigated.

Dimitriev Yu.K. **EFFICIENCY OF LOCAL SELF-DIAGNOSING IN COMPUTING SYSTEMS WITH CIRCULANT DIAGNOSTIC STRUCTURE.** Diagnosing of computing systems (CS) is examined under conditions of the t-multiple faults and test outcomes corresponding to the known PMC-model. A circulant graph with the nodes of degree t is used as the diagnostic structure. In the paper we investigate the opportunity to determine a technical state of each module of the CS using results of testing only those modules which have with it physical connections (local t-diagnosing). Possibility of local diagnosing is investigated under conditions when the rules early entered by us provide completeness of diagnosing only at the expense of labour-consuming additional testing of modules. The conditions of the authentic identification of the CS module state on the basis of redundancy among the test outcomes used at the comparative analysis for the determination of the module state are deduced. It is shown that redundancy in the elements for the comparative analysis reduces level in redundancy on number of the additional tests or even excludes the additional testing fully. Conditions at which redundancy of the elements for the comparative analysis are sufficient for maintenance local t-diagnosability of the system are proved.

Dulkeyt V.I., *Faizullin R.T.*, *Khnikin I.G.* **GLOBAL OPTIMIZATION PROBLEMS ASSOCIATED WITH CRYPTOGRAPHIC ANALYSIS OF ASYMMETRIC CIPHERS.** The aim of this article is to establish relation between well-known problems of cryptographic analysis and global optimization problems which can be associated with SAT representation of cryptographic algorithms where bits of key are part of SAT solution string. The SAT forms for factorization problem, for logarithmic problem and for logarithmic problem on elliptic curves are constructed. For numerical solution, some low relaxation algorithms are adapted. The results of numerical experiments give us more then 50% bits for unknowns in SAT factorization form.

Fomichev V.M. **ON *C*-WIDTH OF FINITE NONCYCLIC GROUPS.** Some properties of finite noncyclic groups cover with maximal cyclic subgroups are investigated. Method to construct group of linear substitutions of degree 2^n which has c-width not less than 2^n is proposed.

Glukhov M.M. **SOME APPLICATIONS OF QUASIGROUPS IN CRYPTOGRAPHY**. It is a survey of the known results related to the applications of quasigroups for constructing authentication codes, ciphers, and one-way functions.

Grishin A.M. **METHODS FOR PROTECTING SPEECH INFORMATION.** Telephone conversations are the major channel of information interaction. The requirements for maintenance of speech confidentiality and for protecting information having the speech nature are growing. In the paper some problems arising in connection with the speech information protection are considered, and some ways for solving them are offered.

Ivaschenko E.A., *Skobelev V.G.* **PRESENTATION OF CRYPTOSYSTEMS VIA POLYBASIC ALGEBRAIC SYSTEM.** A formal model of cryptosystem based on Polybasic Algebraic System is introduced. Basic types of cryptosystems are designed in terms of this formal model.

Kachanov M.A., *Kolegov D.N.* **SECURITY LINUX EXTENSION BASED ON SYSTEM CALLS INTERPOSITION.** The paper reviews some famous Linux security tools based on system calls interception. The LKM model of security policy enforcement for Linux is offered.

Kolegov D.N., *Chernushenko Y.N.* **ABOUT THE CTF – COMPUTER SECURITY COMPETITIONS.** The CTF conception of computer security competitions is described. This approach can be used for effective education on computer security specialities.

Kovalev A.M., *Kozlovskii V.A.*, *Shcherbak V.F.* **INVERSE DYNAMICAL SYSTEMS WITH VARIABLE DIMENSION OF PHASE SPACE IN PROBLEMS OF CRYPTOGRAPHIC INFORMATION TRANSFORMATION.** The paper considers modern algorithms of information protection based on discrete inverse dynamical systems. The effect of dynamical degradation of transformation system is identified. The attraction of trajectories to invariant sets of smaller dimension results in resistance decrease of corresponding algorithms. To compensate for this effect the method of phase space dimension regulation is proposed. For a Lorenz system, the transition from the prototype system to its automaton analogue specified by equations over a finite ring or a Galois field is described. For such an automaton an algorithm for controlling the state space and input space dimensions by means of introduction of certain predicates is described. It is proved that the obtained automaton might be considered us a subautomaton of some infinite automaton over the algebraic extension of the initial Galois field.

Kukartsev A.M., *Popov A.M.*, *Shestakov V.S.* **ABOUT THE DIRECT OPERATIONAL ANALYSIS OF SYMMETRIC CIPHERS.** The concept of cryptoalgorithm complexity as a relative characteristic of security is introduced. The method of direct operational analysis is applied to mathematical research of cryptographic algorithms and their keys. Realization of the method is described.

Lapshin V.V. **CENTRALIZED ANALYSIS OF GEOGRAPHICALLY-DISTRIBUTED NETWORK TRAFFIC**. The paper covers the centralized analysis of geographically-distributed network traffic. Traffic capture techniques, captured traffic delivery,

aggregation, analysis and decision-making are presented. A special GNU/Linux distribution with integrated PF_RING technology is created. It makes possible successful hi-speed (1Gb/s) traffic capture. The captured traffic delivery system consists of two parts: client(s) and server. Both applications use encryption methods to transport captured traffic. The encryption methods are virtually unlimited due to the extensible encryption primitives. By default, a probabilistic stream cryptosystem called libpssc is used. After decryption, all the captured traffic is directed to a processing center where it becomes available for analysis. The processing center allows to plug in at real-time special independent plugins which analyze the traffic according to a criteria. Many plugins can work at once. Pilot implementation details and other results are reported also.

Melent'ev V.A. **FUNCTION OF STRUCTURAL FAULT TOLERANCE AND *d*-LIMITED CONNECTED COMPONENT OF CS GRAPH.** The concept of *d*-limited connected component in computing system graph is introduced. Fault tolerance functions for computer system of ring and completely connected structures are considered. It is shown that they bound the domain of fault tolerance functions for all other structures.

Melent'ev V.A. **SEARCHING (*s*, *t*)-CUTSETS OF COMPUTING SYSTEM GRAPH WITH DIAMETER LIMITING OF CONNECTED COMPONENTS.** An approach to searching the minimum *d*-limited (*s*,*t*)-cutsets in computer system graph is presented. The approach is based on bracket images and projections of the graph.

Muradov R.M., *Kirov V.A*. **ON QUASIGROUPS ARISING FROM PHYSICAL STRUCTURE OF (2, 2) RANK.** In this article the quasigroup giving the physical structure of (2, 2) rank is transferred by isotopy to the quasigroup with the right unity and with identity operation of taking the right inverse element.

Parvatov N.G. **PERFECT SECRET SHARING SCHEMES.** Some results about perfect secret sharing schemes are presented.

Pautov P.A. **THE PROBLEM OF AUTHENTICATION IN THE MULTI-TIER APPLICATIONS.** The paper considers peculiarities of authentication in multi-tier environment and corresponding security problems. The method increasing security of the most popular in modern web-applications two-tier authentication scheme is provided.

Pestunova T.M., *Rodionova Z.V.* **PROCESS MANAGEMENT OF ACCESS RIGHTS DISTRIBUTION ON THE BASIS OF THE BUSINESS-PROCESSES ANALYSIS.** In this article we explore the main aspects of constructing the system that grants access rights to users and manages their changes based on the analysis of business-processes of a company. The management of the companies and the business-processes are one of the most widespread ways to improve activity (reduction of expenses, improvement of service quality, etc.). However many modern organizations describe and optimize business-processes separately from their automatization and protection. Therefore scripts of the information system frequently mismatch business-processes and even can conflict with them. The system of access differentiation creates handicaps in the work of users. One of the first problems solved when the system of access differentiation is created is the dividing the corporate information into categories and the definition of access rights to it for various employees of the company. It is obvious that the information system should be worked out considering requirements of dividing corporate information and definition of access rights that follow from analysis of company's business-processes. In the article there is an example of the description of the business-processes in the eEPC-notation and the construction of the access management model based on the analysis of them.

Potapov V.N. **ARITHMETIC CODING OF MESSAGES USING RANDOM SEQUENCE.** A modification of the arithmetic coding using a random sequence as a secret key is suggested. The method achieves the theoretical bound of text compression not only for ordinary message but for partially defined data as well.

Pozdeyev A.G. **GENERATING THE NORMAL PERIODIC SEQUENCES ON THE BASE OF CYCLICALLY MINIMAL NUMBERS.** A method for generating binary normal periodic sequences is suggested. The sequences are built as a result of joining cycles produced by cyclically minimal numbers depending on the value of a parameter playing the role of the key. In the case of the prime order of the sequences generated, the formulas for computing the parameter length and the number of the sequences produced by the method are given.

Pudovkina M.A. **PROPERTIES OF FEISTEL'S CIPHERS RELATIVE TO TWO WREATH PRODUCTS**. The Feistel scheme is widely used in cryptography. In this paper we investigate properties of Feistel's ciphers relative to two wreath products. We describe weaknesses of such ciphers.

Semenov A.A., *Zaikin O.S.*, *Bespalov D.V.*, *Burov P.S.*, *Hmelnov A.E.* **ANALYSIS OF SOME CRYPTOGRAPHIC PRIMITIVES ON COMPUTER CLUSTERS.** This paper is a continuation of a series of papers devoted to problems of reversing of discrete functions belonging to class that has intersections with different areas of mathematical cybernetics. In particular the functions used in modern cryptosystems as ciphering transformations belong to the given class. In the paper some distinctive moments concerning the solving of considered functions reversing problems on multiprocessing computing systems are discussed.

Skobelev V.G. **ANALYSIS OF ATTACKS ONTO QUANTUM PUBLIC KEY DISTRIBUTION SYSTEM.** The computational security of quantum public key distribution system is investigated under supposition that the cryptanalyst can control probabilities of selection of base vectors intended for qubit's measurement as well as probabilities of simultaneous alternatives of basses of sender and addressee.

Tuzhilin M. **THE ALGEBRAIC IMMUNITY OF BOOLEAN FUNCTIONS.** The recent algebraic attacks have received a lot of attention in cryptographic literature. The algebraic immunity of a Boolean function quantifies its resistance to the standard algebraic attacks of the pseudo-random generators using it as a nonlinear filtering or combining function. This survey contains brief description of algebraic attacks and results have been found concerning the algebraic immunity and its relations with the other cryptographic parameters.

Yegorushkin O.I., *Kalugin-Balashov D.R.*, *Safonov K.V.* **ON A SOLVING OF ALGEBRAIC EQUATIONS SYSTEMS ASSOCIATED WITH CONTEXT-FREE LANGUAGES.** Systems of algebraic equations (polynomial and linear) which appear in context-free languages theory are considered. The solution of the system is a set of formal power series expressing one group of variables trough the other group of variables considered as parameters. It is impossible to directly use the classical elimination processes for a non-commutative ring. The conditions of solubility of a polynomial equations system as well as possibility to make lower the order of a linear equations system are given in the article.

Zolotarev V.V., *Shirkova E.A.* **FOUNDATIONS OF THE METHODS FOR THE BASE EXPERT ANALYSIS OF INFORMATION RISKS.** An attempt is made to analyze the expert estimation procedure of information risk. The main purpose of this analysis is to make clear the fundamental foundations of base methods for information risk estimation. The presented results are received by the set- and graph-theoretical approaches.