

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Приложение

№6

Сентябрь 2013

Свидетельство о регистрации: ПИ №ФС 77-50702
от 17 июля 2012 г.

ТРУДЫ
Всероссийской конференции
«XII Сибирская научная школа-семинар с международным участием
“Компьютерная безопасность и криптография” — SIBECRYPT’13»
(Томск, Парабель, 2–7 сентября 2013 г.)



ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**РЕДАКЦИОННАЯ КОЛЛЕГИЯ ЖУРНАЛА
«ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА»**

Агибалов Г. П., д-р техн. наук, проф. (председатель); Девянин П. Н., д-р техн. наук, проф. (зам. председателя); Парватов Н. Г., д-р физ.-мат. наук, доц. (зам. председателя); Черемушкин А. В., д-р физ.-мат. наук, чл.-корр. Академии криптографии РФ (зам. председателя); Панкратова И. А., канд. физ.-мат. наук, доц. (отв. секретарь); Алексеев В. Б., д-р физ.-мат. наук, проф.; Бандман О. Л., д-р техн. наук, проф.; Быкова В. В., д-р физ.-мат. наук, проф.; Глухов М. М., д-р физ.-мат. наук, академик Академии криптографии РФ; Евдокимов А. А., канд. физ.-мат. наук, проф.; Закревский А. Д., д-р техн. наук, проф., чл.-корр. НАН Беларуси; Колесникова С. И., д-р техн. наук; Костюк Ю. Л., д-р техн. наук, проф.; Логачев О. А., канд. физ.-мат. наук, доц.; Салий В. Н., канд. физ.-мат. наук, проф.; Сафонов К. В., д-р физ.-мат. наук, проф.; Фомичев В. М., д-р физ.-мат. наук, проф.; Чеботарев А. Н., д-р техн. наук, проф.; Шойтов А. М., д-р физ.-мат. наук, чл.-корр. Академии криптографии РФ; Шоломов Л. А., д-р физ.-мат. наук, проф.

Адрес редакции: 634050, г. Томск, пр. Ленина, 36

E-mail: vestnik_pdm@mail.tsu.ru

Всероссийская конференция «XII Сибирская научная школа-семинар с международным участием “Компьютерная безопасность и криптография” — SIBECRYPT’13» проведена Национальным исследовательским Томским государственным университетом в сотрудничестве с Институтом криптографии, связи и информатики со 2 по 6 сентября 2013 г. в г. Томске и с. ПарABELи Томской области при финансовой поддержке РФФИ (грант № 13-07-06036-г).

**Теоретические основы прикладной дискретной математики
Математические методы криптографии
Математические основы компьютерной безопасности и надёжности
вычислительных и управляющих систем
Прикладная теория графов
Математические основы информатики и программирования
Вычислительные методы в дискретной математике**

Редактор *Н. И. Шидловская*

Верстка *И. А. Панкратовой*

Подписано к печати 20.07.2013.

Формат 60 × 84¹/₈. Усл. п. л. 15,7. Уч.-изд. л. 17,6. Тираж 300 экз.

Издательство ТГУ. 634029, Томск, ул. Никитина, 4

СОДЕРЖАНИЕ

Секция 1

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

Аборнев А. В. Разрядно-инъективные преобразования модуля над кольцом Галуа	6
Бондаренко Л. Н. Свойства статистики var на группе перестановок	7
Былков Д. Н. Вторая координатная последовательность линейной рекурренты максимального периода над кольцом \mathbb{Z}_8	9
Волгин А. В. Оценка скорости сходимости в многомерной центральной предель- ной теореме	11
Геут Кр. Л., Титов С. С. О поликвадратичном расширении бинарных полей	12
Заец М. В. Классы полиномиальных и вариационно-координатно полиномиаль- ных функций над кольцом Галуа	13
Коломеец Н. А. Об аффинности булевых функций на подпространствах и их сдвигах	15
Курганский А. Н. Об алгоритмических и топологических свойствах орбит кусочно-аффинных отображений	16
Мироненко О. Л. О статистической независимости произвольной суперпозиции булевых функций	18
Филюзин С. Ю. Верхняя оценка алгебраической иммунности некоторых бент- функций Диллона	19
Фомичев В. М. Эквивалентность примитивных множеств	20
Фролова А. А. Итеративная конструкция APN-функций	24
Черемушкин А. В. К определению степени нелинейности дискретной функции на циклической группе	26
Шоломов Л. А. Экономное представление недоопределённых данных и дизъ- юнктивные коды	27

Секция 2

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

Виткуп В. А. О представлении S-блоков при реализации в блочных шифрах	30
Калужин А. К., Чижов И. В. Алгоритм восстановления открытого текста по шифртексту в криптосистеме Мак-Элиса	32
Карпунин Г. А. О вероятностных характеристиках случайных графов, порожд- даемых алгоритмами поиска коллизий криптографических хэш-функций	33
Катеринский Д. А. Об обратимости конечных автоматов с конечной задержкой	35
Ковалев Д. С. Реализация на ПЛИС симметричного аналога FAPKC	36
Коренева А. М. О блочных шифрах, построенных на основе регистров сдвига с двумя обратными связями	39
Медведев Н. В., Титов С. С. Конструкции идеальных схем разделения секрета	41
Медведева Н. В., Титов С. С. О неминимальных совершенных шифрах	42
Пестунов А. И. О связях между основными понятиями разностного анализа итеративных блочных шифров	44
Чижов И. В., Бородин М. А. Уязвимость криптосистемы Мак-Элиса, постро- енной на основе двоичных кодов Рида — Маллера	48

Секция 3

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ И НАДЁЖНОСТИ ВЫЧИСЛИТЕЛЬНЫХ И УПРАВЛЯЮЩИХ СИСТЕМ

Алехина М. А., Барсукова О. Ю. Об оценках ненадёжности схем при инверсных неисправностях и отказах функциональных элементов.....	50
Анисеня Н. И., Стефанцов Д. А., Торгаева Т. А. Сервис BlackBox для проведения соревнований по защите компьютерной информации Capture The Flag	52
Васин А. В. О базисах с коэффициентом ненадёжности 1.....	56
Девянин П. Н. Корректность правил преобразования состояний системы в рамках мандатной сущностно-ролевой ДП-модели ОС семейства Linux	58
Зайцев Г. Ю., Потапкин А. И., Стефанцов Д. А. Модификация скомпилированных приложений для платформы Android методом аспектно-ориентированного программирования	60
Колегов Д. Н., Ткаченко Н. О., Чернов Д. В. Разработка и реализация мандатных механизмов управления доступом в СУБД MySQL	62
Щерба Е. В., Волков Д. А. Разработка системы обнаружения распределённых сетевых атак типа «отказ в обслуживании»	68

Секция 4

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

Абросимов М. Б., Моденова О. В. О нижней оценке числа дополнительных дуг минимального вершинного 1-расширения ориентации цепи	71
Батуева Ц. Ч.-Д. Свойства генных сетей циркулянтного типа с пороговыми функциями	72
Бондаренко П. П. К вопросу о верхней оценке числа дополнительных рёбер минимальных вершинных расширений цветных циклов	73
Евдокимов А. А., Кочемазов С. Е., Отпущенников И. В., Семенов А. А. Исследование динамических свойств некоторых дискретно-автоматных отображений, заданных случайными графами.....	75
Жаркова А. В. О ветвлении и непосредственных предшественниках состояний в конечной динамической системе всех возможных ориентаций графа	76
Комаров Д. Д. О минимальных рёберных расширениях палм специального вида	78
Корниенко А. С. Деревья функциональных графов для циркулянтов с линейными булевыми функциями в вершинах	80
Кяжин С. Н. О локальной примитивности графов и неотрицательных матриц.....	81
Нажмиденова А. М. Дискретная динамическая система на двойном циркулянте с разными функциями в вершинах	84
Осипов Д. Ю. О T-неприводимых расширениях сверхстройных деревьев	85
Салий В. Н. Об упорядоченном множестве связанных частей многоугольного графа	87
Токарева Н. Н. Простое доказательство сильной регулярности графа Кэли бент-функции	89
Цициашвили Г. Ш., Осипова М. А., Лосев А. С. Асимптотики вероятностей связности пар вершин графа	90

Секция 5

**МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ
И ПРОГРАММИРОВАНИЯ**

Agibalov G. P., Lipsky V. B., Pankratova I. A. Cryptographic extension of Russian programming language	93
Agibalov G. P., Lipsky V. B., Pankratova I. A. Project of hardware implementation of Russian programming language	98
Broslavskiy O. V. AES in LYaPAS	102

Секция 6

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

Анашкина Н. В. О возможности сокращения перебора в алгоритме Балаша	105
Арбузов Д. С., Туктарова Л. И. Сравнительный анализ некоторых алгоритмов распознавания гладких чисел	107
Булавинцев В. Г., Семенов А. А. О GPU-реализации ограниченной версии нехронологического алгоритма DPLL	111
Быкова В. В. Об асимптотике решений рекуррентных соотношений в анализе алгоритмов расщепления для пропозициональной выполнимости	112
Жуков К. Д., Рыбаков А. С. К решению больших систем сравнений	116
Климина А. С. Оптимизация $(p - 1)$ -алгоритма Полларда	118
Кузнецова А. С., Кузнецов А. А., Сафонов К. В. Параллельный алгоритм вычисления функций роста в конечных дупорозданных группах периода 5	119
Поттосин Ю. В., Кардаш С. Н. Конвейеризация комбинационных схем	121
Рябокоть Д. В. Алгоритм поиска запретов булевых функций	123
Семенов А. А. Об эффективном представлении дизъюнктивных нормальных форм диаграммами специального вида	125
Усатюк В. С. Реализация параллельного алгоритма поиска кратчайшего вектора в блочном методе Коркина — Золотарева	130
Черняк Р. И. Распараллеливание алгоритма декодирования стандарта сжатия видеоданных H.265/HEVC	131
Шангин Р. Э. Точный алгоритм для решения одного частного случая задачи Вебера в дискретной постановке	136
СВЕДЕНИЯ ОБ АВТОРАХ	138
АННОТАЦИИ ДОКЛАДОВ	143

Секция 1

**ТЕОРЕТИЧЕСКИЕ ОСНОВЫ
ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ**

УДК 519.7

**РАЗРЯДНО-ИНЪЕКТИВНЫЕ ПРЕОБРАЗОВАНИЯ МОДУЛЯ
НАД КОЛЬЦОМ ГАЛУА¹**

А. В. Аборнев

Предлагается новый способ построения большого класса нелинейных подстановок над кольцом Галуа.

Ключевые слова: *разрядно-инъективная матрица, РИ-матрица, подстановка, кольцо Галуа.*

Предлагается способ построения нелинейных подстановок h на модуле ${}_R R^m$, $m \geq 1$, над кольцом Галуа $R = \text{GR}(q^2, p^2)$, $q = p^r$, представимых с помощью линейных разрядно-инъективных преобразований этого модуля и соответствующих p -адическому разрядному множеству $P = \Gamma(R) = \{a \in R : a^q = a\}$ кольца R .

Каждый элемент $a \in R$ однозначно представляется в виде [1]

$$a = a_0 + pa_1, \quad a_s = \gamma_s(a) \in P, \quad s \in \{0, 1\}.$$

Здесь $\gamma_s: R \rightarrow P$ — разрядные функции в разрядном множестве P . Тогда (P, \oplus, \cdot) — поле с операцией $x \oplus y = \gamma_0(x+y)$. Для матриц $A \in R_{m,n}$ также справедливо разложение $A = A_0 + pA_1$, $A_s = \gamma_s(A) \in P_{m,n}$, $s \in \{0, 1\}$.

А. А. Нечаевым предложен следующий способ построения подстановок. Назовём матрицу K размеров $m \times n$ над кольцом Галуа R *разрядно-инъективной (РИ-матрицей)*, если любая ненулевая строка $\mathbf{a} \in R^m$ однозначно восстанавливается по строке $\gamma_1(\mathbf{a}K) \in P^n$.

Теорема 1. Пусть $G \in P_{m,m}^*$, $U \in R_{m,m}^*$. Тогда матрица

$$K = U(E \mid E + pG)$$

является разрядно-инъективной и отображение $h: R^m \rightarrow R^m$, действующее на произвольной строке $\mathbf{x} \in R^m$ по правилу

$$h(\mathbf{x}) = \mathbf{z}, \quad \text{где } \mathbf{z} = \mathbf{z}_1 + p\mathbf{z}_2 \in R^m, \quad (\mathbf{z}_1 \mid \mathbf{z}_2) = \gamma_1(\mathbf{x}K) \in P^{2m}, \quad (1)$$

является подстановкой.

Для множества подстановок вида $(\Sigma h)^k$, где Σ — регулярное представление группы $(R^m, +)$ в симметрической группе $S(R^m)$, изучаются следующие параметры: показатель 2-транзитивности $d_2(\Sigma h)$ (минимальное k , при котором 2-транзитивно множество $(\Sigma h)^k$) и порождаемая группа [2, 3]. Получены следующие результаты.

¹Работа выполнена при поддержке Академии криптографии РФ.

Теорема 2. Для любой подстановки h вида (1) верно неравенство $d_2(\Sigma h) \geq 4$.

Теорема 3. Пусть $R = \text{GR}(q^2, p^2)$, $m = 1$, $p > 2$. Тогда если разрядное множество $P = \Gamma(R)$ удовлетворяет условию

$$\{\gamma_1(a + e) \ominus \gamma_1(b + e) : a, b \in P\} = P,$$

то для любой подстановки h вида (1) справедливо равенство $d_2(\Sigma h) = 4$.

Если при этом $R = \mathbb{Z}_{p^2}$, то группа $\langle \Sigma h \rangle$ содержит знакопеременную группу A_{p^2} .

Теорема 4. Пусть $R = \text{GR}(q^2, 4)$, $m > 1$. Тогда если все миноры матрицы U_0 в (1) ненулевые, то для подстановки h из (1) справедливо равенство $d_2(\Sigma h) = 4$.

Автор выражает глубокую благодарность профессору А. А. Нечаеву за постановку задачи и внимание к проводимым исследованиям.

ЛИТЕРАТУРА

1. Кузьмин А. С., Нечаев А. А. Линейные рекуррентные последовательности над кольцами Галуа // Алгебра и Логика. 1995. Т. 34. № 2. С. 169–189.
2. Глухов М. М. О 2-транзитивности произведения регулярных групп подстановок // Труды по дискретной математике. М.: Физико-математическая литература, 2000. С. 37–52.
3. Глухов М. М., Елизаров В. П., Нечаев А. А. Алгебра. Т 2. М.: Гелиос АРВ, 2003. 416 с.

УДК 519.1

СВОЙСТВА СТАТИСТИКИ VAR НА ГРУППЕ ПЕРЕСТАНОВОК¹

Л. Н. Бондаренко

Рассматриваются некоторые свойства статистики var, определяющей число различных символов слова, полученного поэлементным сложением по mod n перестановки степени n с фиксированной перестановкой — ключом.

Ключевые слова: перестановка, статистика, производящий многочлен, перманент, циркулянт.

Ряд статистик на симметрической группе S_n перестановок σ , где $\sigma = \sigma_1 \dots \sigma_n$ — слово над алфавитом $\{1, \dots, n\}$, рассматривался в [1].

В криптографии применяется биекция $\text{vp} : \sigma \mapsto \tau$ перестановки $\sigma \in S_n$ на слово $\tau = \tau_1 \dots \tau_n \in T_n$, определяемая фиксированным ключом $\varkappa = \varkappa_1 \dots \varkappa_n \in S_n$. Она отображает $\sigma \in S_n$ на слово $\tau \in T_n$ по правилу $\tau = \sigma \oplus \varkappa$, где $\tau_i = \sigma_i + \varkappa_i \pmod{n}$, $i = 1, \dots, n$, а τ_i — наименьший положительный вычет, и индуцирует статистику $\text{var}(\sigma, \varkappa) = \text{card}\{\tau_1, \dots, \tau_n\}$ — число различных символов слова $\tau = \text{vp}(\sigma)$.

Так как для любого ключа $\varkappa \in S_n$ статистика var имеет производящий многочлен

$$V_n(t) = \sum_{k=1}^n V_{n,k} t^k = \sum_{\sigma \in S_n} t^{\text{var}(\sigma, \varkappa)},$$

то в качестве ключа удобно использовать перестановку $\nu = \nu_1 \dots \nu_n \in S_n$, где $\nu_i = n - i + 1$, $i = 1, \dots, n$. Многочлен $V_n(t)$ не изменяется и при замене перестановок σ на обратные $\sigma^{-1} \in S_n$.

По определению статистики var многочлен $V_n(t)$ имеет коэффициенты $V_{n,k} \geq 0$, а их нахождение уже при сравнительно небольших n является трудной задачей. Вычисление даёт $V_1(t) = V_2(t)/2 = t$, $V_3(t)/3 = t + t^3$, $V_4(t)/4 = t + t^2 + 4t^3$, $V_5(t)/5 = t + 20t^3 + 3t^5$.

¹Работа поддержана грантом РФФИ, проект № 11-01-00212а.

Теорема 1. Справедливы следующие свойства коэффициентов многочлена $V_n(t)$: а) $n|V_{n,k}$; б) при чётном n все $V_{n,k} > 0$, кроме $V_{n,n} = 0$; в) при нечётном составном n все $V_{n,k} > 0$, кроме $V_{n,n-1} = 0$, а для простого n , кроме того, $V_{n,2} = 0$.

Доказательство теоремы 1 основано на применении перманента циркулянта

$$\text{circ}(z_1, \dots, z_n) = \begin{pmatrix} z_1 & z_2 & \dots & z_n \\ z_n & z_1 & \dots & z_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ z_2 & z_3 & \dots & z_1 \end{pmatrix},$$

порожденного вектором (z_1, \dots, z_n) . Каждый член $\text{per circ}(z_1, \dots, z_n)$ имеет вид $z_1^{m_1} \dots z_n^{m_n}$, причём $0 \leq m_i \leq n$, а сумма $\sum_{i=1}^n im_i$ кратна n . Число различных членов $\text{per circ}(z_1, \dots, z_n)$ равно [2]

$$\frac{1}{n} \sum_{d|n} \binom{2d-1}{d} \varphi\left(\frac{n}{d}\right),$$

где $\varphi(d)$ — функция Эйлера числа d . Рассматриваемый $\text{per circ}(z_1, \dots, z_n)$ преобразуется в многочлен $V_n(t)$ при замене каждого m_i в выражении $z_1^{m_1} \dots z_n^{m_n}$ на $\text{sign}(m_i)$ и отождествлении $z_1 = \dots = z_n = t$.

Задание равномерной меры на множестве S_n и использование нормировки многочлена $V_n(t)$, заключающейся в его делении на число $V_n(1)$, позволяет вместо последовательности коэффициентов $\{V_{n,k}\}_{k=1}^n$ рассматривать распределение, отвечающее случайной величине X_n .

Теорема 2. Математическое ожидание

$$E(X_n) = n \left(1 - \frac{D_n}{n!}\right) \sim n \left(1 - \frac{1}{e}\right),$$

где D_n — число беспорядков на множестве S_n , т. е. число перестановок $\sigma \in S_n$, не имеющих неподвижных элементов.

Для доказательства теоремы 2 с помощью свойств чисел D_n [3] устанавливается, что при фиксации символа $i \in \{1, \dots, n\}$ количество слов $\tau \in T_n$, не содержащих этого символа, равно D_n , и применяются свойства математического ожидания. Асимптотика $E(X_n)$ следует из соотношения $D_n \sim n! e^{-1}$.

Записывая $V_n(t) = V_n^+(u) + tV_n^-(u)$, где $u = t^2$, с помощью теоремы 1 получаем, что при $n \geq 6$ $V_n^+(u) = u^m U_n^+(u)$, причём $m = 2$ для простого n и $m = 1$ для составного n . Вычисления показывают, что при $n \geq 6$ все корни многочленов $U_n^+(u)$ и $V_n^-(u)$, исключая $U_7^+(u) = 1$, различны и отрицательны.

Доказательство этого свойства явилось бы существенным моментом для нахождения асимптотики функции распределения случайной величины X_n .

ЛИТЕРАТУРА

1. *Фоата Д.* Распределения типа Эйлера и Макмагона на группе перестановок // Проблемы комбинаторного анализа: сб. статей. М.: Мир, 1980. С. 120–141.
2. *Brualdi R. A. and Newman M.* An enumeration problem for a congruence equation // J. Research National Bureau Standards. Math. Sci. 1970. V. 74B. No. 1. P. 37–40.
3. *Риордан Дж.* Введение в комбинаторный анализ. М.: ИЛ, 1963.

УДК 519.7

ВТОРАЯ КООРДИНАТНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ЛИНЕЙНОЙ РЕКУРРЕНТЫ МАКСИМАЛЬНОГО ПЕРИОДА НАД КОЛЬЦОМ \mathbb{Z}_8 ¹

Д. Н. Былков

Описывается аналитическое строение второй координатной последовательности линейной рекурренты над кольцом \mathbb{Z}_8 . Уточняется нижняя оценка ранга (линейной сложности), строятся классы многочленов и рекуррент максимального периода, у которых достигается максимально возможный ранг.

Ключевые слова: линейная рекуррентная последовательность над кольцом, координатная последовательность, ранг, аналитическое строение.

Пусть $R = \mathbb{Z}_8$, $F(x) \in R[x]$ — многочлен максимального периода (МП-многочлен) степени m , т. е. унитарный (со старшим коэффициентом e) многочлен степени m с максимально возможным при данном m периодом $T(F) = 2^2(2^m - 1)$ [1]. Обозначим через $L_R(F)$ множество всех линейных рекуррент над R с характеристическим многочленом $F(x)$, а через $L_R(F)^*$ — подмножество линейных рекуррент $u \in L_R(F)$ периода $T(u) = T(F)$, т. е. линейных рекуррентных последовательностей максимального периода (МП ЛРП).

Подмножество $K = \{k_0, k_1\} \subset R$ назовем координатным множеством кольца R (см., например, [1]), если справедливы соотношения $k_0 \in 2R$, $k_1 \in R^*$. Примером координатного множества кольца R является двоичное координатное множество $K = \{0, 1\}$. Пусть $a \in R$, хорошо известно разложение $a = a_0 + 2a_1 + 2^2a_2$, $a_i = \gamma_i^K(a) \in K$, $i = 0, 1, 2$, называемое разложением элемента a в координатном множестве K . Элемент a_2 будем называть старшей координатой элемента a в координатном множестве K . На множестве K можно задать структуру поля: $a \otimes b = \gamma_0^K(a * b)$, $* \in \{+, \cdot\}$.

Пусть $F(x) \in R[x]$ — унитарный многочлен Галуа (т. е. неприводимый по модулю 2) степени $m \geq 5$. Рассмотрим последовательность $u_2 = \gamma_2^K(u)$, полученную разложением знаков ЛРП u в координатном множестве K : $u_2(i) = \gamma_2^K(u(i))$, $i \in \mathbb{N}_0$.

Для случая, когда многочлен $F(x)$ является отмеченным ($T(F) = 2^m - 1$), в работе [2] найдено разложение второй двоичной координатной последовательности u_2 в сумму биномиальных последовательностей.

В настоящей работе найдено биномиальное разложение второй координатной последовательности $\gamma_2^K(u)$ при произвольном выборе координатного множества K . Пусть θ — корень многочлена $F(x)$ в расширении S кольца R , $\Gamma(S) = \{a^{2^{3m}} = a : a \in S\}$ — координатное множество Тейхмюллера. Тогда знак ЛРП u представляется в виде $u(i) = \text{Tr}_R^S(\xi \theta^i)$, $\xi \in \Gamma(S)$. Пусть также $\xi = \xi_0 + 2\xi_1 + 4\xi_2$, $\theta = \theta_0 + 2\theta_1 + 4\theta_2$ — разложения элементов ξ и θ в координатном множестве $\Gamma(S)$. Тогда справедлива

Теорема 1. Для знака последовательности u_2 справедливо равенство

$$u_2(i) = \binom{i}{2} \text{tr}(\xi_0(\nu + \nu^2)w^i) \oplus i s_2(i) \oplus s_1(i) \oplus g^K(i \text{tr}(\xi_0 \nu w^i) + \text{tr}(\xi_1 w^i) + \sigma_2(\xi_0 w^i), \text{tr}(\xi_0 \nu w^i)),$$

¹Работа выполнена при поддержке гранта президента РФ № НШ-6260.2012.10.

где $\nu = \theta_1/\theta_0$, $\text{tr}(x) = \text{tr}_{\{0,1\}}^{\Gamma(S)}(x)$, многочлен g^K определяется лишь выбором K ,

$$\begin{aligned} s_2(i) &= \text{tr}(\xi_0 \nu w^i) \sigma_2(\xi_0 w^i) \oplus \sigma_2(\xi_0 \nu w^i) \oplus \text{tr}(\xi_0 \nu w^i) \text{tr}(\xi_1 w^i) \oplus \text{tr}(\xi_0 \theta_2 w^{i-1}) \oplus \text{tr}(\xi_1 \nu w^i); \\ s_1(i) &= \sigma_4(\xi_0 w^i) \oplus \sigma_2(\xi_1 w^i) \oplus \text{tr}((\xi_0 \oplus \xi_1) w^i) \sigma_2(\xi_0 w^i) \oplus \text{tr}(\xi_0 w^i) \sigma_3(\xi_0 w^i) \oplus \text{tr}(\xi_2 w^i); \\ \sigma_2(x) &= \sum_{0 \leq i < j \leq m-1} x^{2^i+2^j}, \sigma_3(x) = \sum_{0 \leq i < j < k \leq m-1} x^{2^i+2^j+2^k}, \sigma_4(x) = \sum_{0 \leq i < j < k < s \leq m-1} x^{2^i+2^j+2^k+2^s}. \end{aligned}$$

Доказательство теоремы 1 получено при помощи метода *исключения младших координат*, разработанного В. Л. Куракиным [3].

На основе результата теоремы 1 удалось уточнить известные оценки линейной сложности (ранга) второй двоичной координатной последовательности и получить оценки ранга в случае произвольного координатного множества.

В работе [1] получены нижние оценки ранга двоичной координатной последовательности $u_2 = \gamma_2^{\{0,1\}}(u)$ в случае, когда $F(x)$ — МП-многочлен. В частности, описаны ограничения на выбор МП-многочлена $F(x)$, при которых для произвольной ЛРП $u \in L_R(F)^*$ справедливы неравенства

$$3m + 2 \binom{m}{3} + \binom{m}{4} \leq \text{rk } u_2 \leq 3m + 2 \binom{m}{3} + 2 \binom{m}{2} + \binom{m}{4}, \quad (1)$$

причём верхняя оценка в неравенстве (1) справедлива для любого МП-многочлена.

Справедлива

Теорема 2.

а) Если гарантированный ранг системы элементов $\{\nu, \nu^2, \dots, \nu^{2^{m-1}}\}$ не меньше 3 или $\text{tr}(\nu) = e$, то для произвольной МП ЛРП $u \in L_R(F)^*$ справедливо неравенство

$$\text{rk}(\gamma_2^K(u)) \geq 3m + 2 \binom{m}{3} + \binom{m}{4}. \quad (2)$$

б) Если элемент ν образует нормальный самодвойственный базис $\Gamma(S)$ над $\{0, 1\}$ и элемент $\xi \in S$ имеет вид $\xi = (3 + 4c)\theta^t$ для некоторых $c \in S$, $t \in \mathbb{N}$, то для ЛРП $u \in L_R(F)^*$ вида $u(i) = \text{Tr}(\xi \theta^i)$ верхнее неравенство из (1) обращается в равенство.

Так как условию $\text{tr}(\nu) = e$ удовлетворяет половина всех МП-многочленов, то нижняя оценка (2) справедлива не менее чем для половины МП-многочленов и произвольного координатного множества K .

ЛИТЕРАТУРА

1. Kurakin V. L., Kuzmin A. S., Mikhalev A. V., and Nechaev A. A. Linear Recurring Sequences over Rings and Modules // J. Math. Sci. (New York). 1995. V. 76. No. 6. P. 2793–2915.
2. Hellesteth T. and Martinsen H. M. Binary sequences of period $2^m - 1$ with large linear complexity // Information and Computation. 1999. V. 151. P. 73–91.
3. Куракин В. Л. Первая координатная последовательность линейной рекурренты максимального периода над кольцом Галуа // Дискретная математика. 1994. Т. 6. № 2. С. 88–100.

УДК 519.21

ОЦЕНКА СКОРОСТИ СХОДИМОСТИ В МНОГОМЕРНОЙ ЦЕНТРАЛЬНОЙ ПРЕДЕЛЬНОЙ ТЕОРЕМЕ

А. В. Волгин

Известна оценка скорости многомерной нормальной аппроксимации для сумм локально зависимых случайных векторов, которая неявным образом зависит от размерности суммируемых векторов. Приводится явный вид зависимости оценки от размерности.

Ключевые слова: многомерная центральная предельная теорема, скорость сходимости, локально зависимые случайные векторы.

В [1] рассматривается оценка скорости нормальной аппроксимации суммы локально зависимых случайных векторов $W = \sum_{i=1}^n X_i$, $X_i \in \mathbb{R}^d$. Предполагается, что слагаемые ограничены так, что $|X_i| \leq B$, $i = 1, \dots, n$, где через $|\cdot|$ обозначается сумма абсолютных значений координат вектора (или элементов матрицы), B — некоторая константа. Под локальной зависимостью подразумевается существование декомпозиций

$$W = U_i + V_i, \quad W = R_i + T_i, \quad i = 1, \dots, n, \quad (1)$$

таких, что $|U_i| \leq A_1$, $|R_i| \leq A_2$, $i = 1, \dots, n$, для некоторых констант $A_1 \leq A_2$. В качестве меры аппроксимации нормальным распределением рассматривается величина

$$\Delta = \sup_{A \in \mathcal{A}} |\mathbb{P}(W \in A) - \mathbb{P}(Z \in A)|,$$

где Z — случайный вектор, имеющий d -мерное стандартное нормальное распределение; \mathcal{A} — класс всех измеримых выпуклых подмножеств \mathbb{R}^d .

Теорема 1 [1]. Пусть $W = \sum_{i=1}^n X_i$ представляется в виде декомпозиций (1), I — единичная матрица размера $d \times d$ над полем \mathbb{R} ,

$$\chi_1 = \sum_{i=1}^n \mathbb{E}|\mathbb{E}(X_i|V_i)|, \quad \chi_2 = \sum_{i=1}^n \mathbb{E}|\mathbb{E}(X_i U_i^T) - \mathbb{E}(X_i U_i^T | T_i)|, \quad \chi_3 = \left| I - \sum_{i=1}^n \mathbb{E}(X_i U_i^T) \right|.$$

Тогда существует константа c , зависящая от размерности векторов d , такая, что

$$\Delta \leq c [aA_2 + naA_1A_2B (|\ln A_2B| + \ln n) + \chi_1 + (|\ln A_1B| + \ln n) (\chi_2 + \chi_3)], \quad \text{где } a \leq \sqrt{2d}.$$

Оценка величины Δ в теореме 1 приводится в условиях фиксированной размерности векторов $d \equiv \text{const} \in \mathbb{N}$ и роста числа слагаемых $n \rightarrow \infty$. В данном случае явный вид зависимости величины c от размерности d не имеет значения.

Рассматривается задача оценки величины Δ при условии одновременного роста размерности векторов и числа слагаемых $d, n \rightarrow \infty$. Предлагается подход, который основан на уточнении явного вида зависимости величины Δ от размерности d .

Теорема 2. В условиях теоремы 1

$$\Delta \leq 3(2\pi)^{d/2} aA_2 + d(2\pi)^{d/2} naA_1A_2B (\ln |17(2\pi)^{d/2} A_1A_2B| + \ln n) + 2d\chi_1 + \\ + d(d+1) (\ln |17(2\pi)^{d/2} A_1A_2B| + \ln n) (\chi_2 + \chi_3), \quad a \leq \sqrt{2d}.$$

ЛИТЕРАТУРА

1. Rinott Y. and Rotar V. A multivariate CLT for local dependence with $n^{-1/2} \log n$ rate and applications to multivariate graph related statistics // J. Multivariate Analysis. 1996. V. 56. P. 333–350.

УДК 512.62

О ПОЛИКВАДРАТИЧНОМ РАСШИРЕНИИ БИНАРНЫХ ПОЛЕЙ

Кр. Л. Геут, С. С. Титов

Работа посвящена генерации неприводимых многочленов степени 2^n посредством поликватратичного расширения поля над $\text{GF}(2)$. Построено полное бинарное дерево неприводимых многочленов, рассмотрены свойства этого расширения.

Ключевые слова: неприводимый многочлен, поликватратичное расширение, след многочлена.

Неприводимые многочлены активно применяются при передаче информации по каналам связи в виде битовых строк, помехоустойчивом кодировании, работе конечных автоматов, стандартах защиты информации [1]. Использование свойств неприводимых многочленов позволяет максимизировать эффективность компьютерной реализации арифметики в конечных полях.

Рассмотрим уравнение $x^2 + x = z$ в поле $\text{GF}(2^m)$, где z — корень неприводимого многочлена f над $\text{GF}(2)$ степени m . Если x не лежит в этом же поле $\text{GF}(2^m)$, а лежит в расширении $\text{GF}(2^{2m})$ этого поля, то след $\text{Tr}(f) = 1$ [2] и x — корень неприводимого многочлена F степени $2m$ [3–5]. Многочлен F получается из f посредством так называемой операции A [6]: $F(X) = f(X^2 + X)$. Многочлен F неприводимый и периодический (с периодом равным единице), т. е. $F(X + 1) = F(X)$.

Если же x лежит в том же поле $\text{GF}(2^m)$, то многочлен, полученный с помощью операции A из многочлена f , приводим, $F(X) = p(X)q(X)$, $\text{Tr}(f) = 0$, $\deg p = \deg q = m$, и x — корень одного из этих двух неприводимых многочленов p, q , связанных соотношением сдвига: $p(X + 1) = q(X)$ [6].

Если $m = 2^k$ и $\text{Tr}(x) = 1$, то после поэтапного применения операции A можно построить полное бинарное дерево, ветви которого символизируют применение операции A , а вершины — многочлены, полученные с помощью этой операции [6]. Если представить шаг (т. е. применение операции A) как уровень, то можно заметить, что вершина с неприводимым многочленом степени 2^{n+1} появляется только после прохождения всех уровней с неприводимыми многочленами степени 2^n , причём число этих уровней-шагов равно 2^{n-1} . Приводимость многочлена, полученного с помощью операции A , зависит от следа многочлена, к которому была применена эта операция.

Многочлены степени 2^n со значением следа $\text{Tr} = 1$ всегда лежат на нижнем уровне, так как после применения операции A они дают неприводимый периодический многочлен степени 2^{n+1} , а на остальных уровнях лежат многочлены с нулевым значением следа $\text{Tr} = 0$.

Симметричные (самовозвратные) многочлены (т. е. многочлены, у которых коэффициенты симметричны относительно центрального бита) степени 2^{n+1} с нулевыми значениями следа и антиследа ($\text{Tr} = \text{Tr}^{-1} = 0$) получаются из многочленов степени 2^n со значениями $\text{Tr} = 0$, $\text{Tr}^{-1} = 1$, а многочлены степени 2^{n+1} со значениями $\text{Tr} = \text{Tr}^{-1} = 1$ — из аналогичных многочленов степени 2^n со значениями $\text{Tr} = \text{Tr}^{-1} = 1$,

причём в одной ветви может встретиться только один симметричный многочлен одной степени.

В результате исследования поликватратичного расширения полей посредством операции A получены следующие свойства этого расширения.

1. Посредством поликватратичного расширения можно вычислять характеристический многочлен элемента не только из расширенного поля, но и из расширяемого, т. е. движение по дереву возможно как вверх, так и вниз. Для того чтобы «опуститься» по дереву вниз, необходимо применить операцию A , а чтобы «подняться» по дереву вверх — выполнить обратную операцию $\Lambda = A^{-1}$.

2. Для вычисления расширений необходимо вычислить относительный след корня и записать уравнение, задающееся неприводимым многочленом, коэффициенты которого вычисляются в явном виде [3, 4].

Теорема 1. Если $h(x) = z$, $\deg f = n$, $f(z) = 0$, $\deg g = 2n$, $g(z) = 0$, то $\text{Tr}(z) = \text{Tr}(x)$, где $h(x) = x + x^{-1}$ равен относительному следу элемента x .

Теорема 2. Если в поле $\text{GF}(2^m)$, $m > 1$, z — корень симметричного многочлена f , то однозначно определён периодический многочлен g , где $y = 1/(z + 1)$ — его корень. И наоборот, если g — периодический, то f — симметричный, где $z = 1/y + 1$.

Таким образом, с помощью операции A построено полное бинарное дерево неприводимых многочленов степени 2^n . Изученные в работе свойства такого поликватратичного расширения значительно упрощают процедуру генерации многочленов и дают возможность избежать полного перебора при поиске многочленов больших степеней, что имеет особое значение для криптографии и теории кодирования.

ЛИТЕРАТУРА

1. Информационные технологии и безопасность алгоритмы разделения секрета. Предварительный государственный стандарт республики Беларусь СТБ П 34.101.44. Минск: Госстандарт, 2011.
2. Глушко Кр. Л., Титов С. С. Арифметический алгоритм решения квадратных уравнений в конечных полях характеристики два // Доклады ТУСУРа. 2012. № 1(25). Ч. 2. С. 148–152.
3. Болотов А. А., Гашков С. Б., Фролов А. Б. Элементарное введение в эллиптическую криптографию: алгебраические и алгоритмические основы. М.: КомКнига, 2006.
4. Болотов А. А., Гашков С. Б., Фролов А. Б. Элементарное введение в эллиптическую криптографию: Протоколы криптографии на эллиптических кривых. М.: КомКнига, 2006.
5. Лидл Р., Нидеррайтер Г. Конечные поля. М.: Мир, 1988.
6. Геут Кр. Л., Титов С. С. О свойствах поликватратичных расширений бинарных полей // Проблемы теоретической и прикладной математики: Труды 44-й Всерос. молодежной конф. Екатеринбург: УрО РАН, 2013. С. 17–19.

УДК 512.55

КЛАССЫ ПОЛИНОМИАЛЬНЫХ И ВАРИАЦИОННО-КООРДИНАТНО ПОЛИНОМИАЛЬНЫХ ФУНКЦИЙ НАД КОЛЬЦОМ ГАЛУА

М. В. Заец

Рассматривается новый класс функций над кольцом Галуа $R = \text{GR}(q^m, p^m)$, получивший название класса функций с вариационно-координатной полиномиальностью (ВКП-функций). Рассматривается соотношение между данным классом и

классом полиномиальных функций над R , даётся верхняя оценка его мощности, а также достаточные условия отсутствия полиномиального представления ВКП-функции.

Ключевые слова: полиномиальные функции, кольцо Галуа, координатное множество, ВКП-функции.

Кольцом Галуа называется конечное коммутативное локальное кольцо $R = \text{GR}(q^m, p^m)$, нильрадикал $J(R)$ которого имеет вид pR , где $p = \text{char } \bar{R}$ и $\bar{R} = R/J(R) = \text{GF}(q)$ — поле вычетов данного кольца [1]. При этом $\text{char } R = p^m$, $|R| = q^m$ и $m = \text{ind } J(R)$, $m \in \mathbb{N}$, — индекс нильпотентности нильрадикала $J(R)$. Подмножество $\mathcal{B} = \{b_0 = 0, \dots, b_{q-1}\} \subseteq R$ называется координатным множеством кольца R , если его элементы образуют полную систему вычетов по нильрадикалу $J(R)$. В таком случае любой элемент $a \in R$ однозначно представляется в виде

$$a = a^{(0)} + p \cdot a^{(1)} + \dots + p^{m-1} \cdot a^{(m-1)}, \quad a^{(i)} \in \mathcal{B}, \quad i = 0, \dots, m-1,$$

называемом разложением элемента a в координатном множестве \mathcal{B} . Функции $\gamma_i^{\mathcal{B}}: R \rightarrow \mathcal{B}$, определяемые по правилу $\gamma_i^{\mathcal{B}}(a) = a^{(i)}$, $i = 0, \dots, m-1$, называются координатными функциями в координатном множестве \mathcal{B} , а элементы $a^{(i)} = \gamma_i^{\mathcal{B}}(a)$ — координатами элемента a в координатном множестве \mathcal{B} .

Обозначим через $\mathcal{P}_R(n)$ класс всех полиномиальных функций от n переменных над кольцом Галуа $R = \text{GR}(q^m, p^m)$. Следующее определение и результаты обобщают полученные ранее в [2] для случая примарного кольца вычетов \mathbb{Z}_{2^m} .

Определение 1. Функцию $f(x): R^n \rightarrow R$, $R = \text{GR}(q^m, p^m)$, $m > 1$, назовём *ВКП-функцией* в координатном множестве \mathcal{B} , если для любого $i \in \{0, \dots, m-1\}$ существует полиномиальная функция $p_i(x) \in \mathcal{P}_R(n)$, такая, что $\gamma_i^{\mathcal{B}}(f(\alpha)) = \gamma_i^{\mathcal{B}}(p_i(\alpha))$ при всех $\alpha \in R^n$. При этом многочлен $p_i(x)$, $i = 0, \dots, m-1$, будем называть i -м координатным многочленом функции $f(x)$.

Класс всех ВКП-функций от n переменных над кольцом R в координатном множестве \mathcal{B} обозначим через $\mathcal{CP}_R^{\mathcal{B}}(n)$. Следующая теорема устанавливает соотношение между введённым классом и классом полиномиальных функций над тем же кольцом.

Теорема 1. Справедливы утверждения:

- 1) если $R = \text{GR}(q^2, p^2)$, то $\mathcal{P}_R(n) = \mathcal{CP}_R^{\mathcal{B}}(n)$;
- 2) если $R = \text{GR}(q^m, p^m)$, $m \geq 3$, то $\mathcal{P}_R(n) \subsetneq \mathcal{CP}_R^{\mathcal{B}}(n)$.

Пусть $f(x) \in R[x]$. Обозначим $\text{grad } f(x) = \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)$, где $\frac{\partial f}{\partial x_i}(x)$ — формальная частная производная многочлена $f(x)$ по переменной x_i , $i = 1, \dots, n$. Приведём достаточное условие того, что при $m \geq 3$ ВКП-функция не имеет полиномиального представления над R .

Теорема 2. Пусть $f(x) \in \mathcal{CP}_R^{\mathcal{B}}(n)$, $m \geq 3$ и для координатных многочленов $p_i(x)$, $p_j(x)$, $i, j \in \{1, \dots, m-1\}$, $i \neq j$, существует $\alpha \in \mathcal{B}^n$, такое, что

$$\text{grad } p_i(\alpha) \not\equiv \text{grad } p_j(\alpha) \pmod{J(R)}.$$

Тогда $f(x) \notin \mathcal{P}_R(n)$.

Теорема 3. Справедлива следующая оценка мощности класса $\mathcal{CP}_R^{\mathcal{B}}(n)$:

$$|\mathcal{CP}_R^{\mathcal{B}}(n)| \leq q^{q^n + (m-1)n \cdot q^n + q^n \cdot (q^{n(m-1)} - 1)/(q^n - 1)},$$

при этом если $m = 2$, то в неравенстве достигается равенство.

Класс ВКП-функций во многом обобщает класс полиномиальных функций. В частности, можно показать, что системы уравнений, левые части которых являются такими функциями, могут быть решены методом покоординатной линеаризации, предложенным в работе [3] для полиномиальных функций над кольцом Галуа — Эйзенштейна.

ЛИТЕРАТУРА

1. *Елизаров В. П.* Конечные кольца. М.: Гелиос-АРВ, 2006.
2. *Заец М. В., Никонов В. Г., Шшиков А. Б.* Функции с вариационно-координатной полиномиальностью и их свойства // Открытое образование. 2012. № 3. С. 57–61.
3. *Михайлов Д. А., Нечаев А. А.* Решение системы полиномиальных уравнений над кольцом Галуа — Эйзенштейна с помощью канонической системы образующих полиномиального идеала // Дискретная математика. 2004. № 1. Вып. 1. С. 21–51.

УДК 519.7

ОБ АФФИННОСТИ БУЛЕВЫХ ФУНКЦИЙ НА ПОДПРОСТРАНСТВАХ И ИХ СДВИГАХ¹

Н. А. Коломеец

Пусть f — булева функция от n переменных и для любого аффинного подпространства L размерности $\lceil n/2 \rceil$ функция f аффинна на L тогда и только тогда, когда f аффинна на любом сдвиге L . Доказано, что тогда либо степень f не превышает 2, либо не существует ни одного аффинного подпространства размерности $\lceil n/2 \rceil$, на котором f аффинна.

Ключевые слова: булевы функции, бент-функции, квадратичные функции.

Рассматривается свойство булевых функций, связанное с их аффинностью на аффинных подпространствах.

Введём необходимые определения. Отображение $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$ называется *булевой функцией* от n переменных. *Алгебраической степенью* или просто *степенью* булевой функции называется степень её алгебраической нормальной формы (полинома Жегалкина). Булева функция называется *аффинной*, если её алгебраическая степень не больше 1, и *квадратичной*, если её степень равна 2. Множество $U \subseteq \mathbb{Z}_2^n$ называется *аффинным подпространством*, если $U = a \oplus L$, где $a \in \mathbb{Z}_2^n$ и L является линейным подпространством в \mathbb{Z}_2^n . Будем называть U сдвигом подпространства L . Через Ind_D обозначим характеристическую функцию множества $D \subseteq \mathbb{Z}_2^n$. Через $\langle u, v \rangle$ обозначим скалярное произведение векторов u и v . Булева функция f от n переменных *аффинна на множестве* $D \subseteq \mathbb{Z}_2^n$, если существуют $a \in \mathbb{Z}_2^n$, $c \in \mathbb{Z}_2$, такие, что для любого $x \in D$ верно $f(x) = \langle a, x \rangle \oplus c$. Под *расстоянием* между двумя булевыми функциями подразумевается *расстояние Хэмминга* между их векторами значений.

Все квадратичные булевы функции обладают следующим свойством.

Утверждение 1. Пусть f — квадратичная булева функция от n переменных. Тогда для любого аффинного подпространства L функция f аффинна на L , если и только если f аффинна на любом сдвиге L .

Доказательство утверждения следует из неравенства $\deg(f(x) \oplus f(x \oplus s)) \leq 1$, верного для любого $s \in \mathbb{Z}_2^n$.

¹Исследование выполнено при поддержке РФФИ (проект № 12-01-31097).

Отметим, что не для всех квадратичных функций существует хотя бы одно аффинное подпространство размерности больше чем $\lceil n/2 \rceil$, на котором функция аффинна. Например, если f является бент-функцией (n чётно), то не существует подпространств размерности $n/2 + 1$, на которых f аффинна. Бент-функция — это булева функция от чётного числа переменных, максимально удаленная от множества всех аффинных функций. Понятие бент-функций ввел О. Ротхаус [1]. Бент-функции представляют интерес в криптографии и теории кодирования, поскольку имеют в этих областях множество различных приложений. Тем не менее до сих пор существует большое количество нерешённых проблем, связанных с бент-функциями [2].

Внесём ограничение на размерность подпространств в условие утверждения 1.

Утверждение 2. Пусть f — квадратичная булева функция от n переменных. Тогда для любого аффинного подпространства L размерности $\lceil n/2 \rceil$ функция f аффинна на L , если и только если f аффинна на любом сдвиге L .

Если f является бент-функцией, то по подпространствам размерности $n/2$, на которых она аффинна, можно строить другие бент-функции.

Утверждение 3 [3]. Пусть f — бент-функция от $2k$ переменных и $L \subseteq \mathbb{Z}_2^{2k}$, $|L| = 2^k$. Тогда $f(x) \oplus \text{Ind}_L(x)$ является бент-функцией тогда и только тогда, когда L является аффинным подпространством и f на нём аффинна.

Таким образом, существует взаимно однозначное соответствие между бент-функциями на расстоянии 2^k от f и аффинными подпространствами, на которых f аффинна.

Следующая теорема показывает, для каких функций, помимо квадратичных, справедливо утверждение 2.

Теорема 1. Пусть f — булева функция от n переменных и для любого аффинного подпространства L размерности $\lceil n/2 \rceil$ функция f аффинна на L , если и только если f аффинна на любом сдвиге L . Тогда либо $\deg f \leq 2$, либо не существует ни одного аффинного подпространства размерности $\lceil n/2 \rceil$, на котором функция f аффинна.

ЛИТЕРАТУРА

1. Rothaus O. On bent functions // J. Combin. Theory. Ser. A. 1976. V. 20. No. 3. P. 300–305.
2. Токарева Н. Н. Нелинейные булевы функции: бент-функции и их обобщения. Saarbrücken: LAP LAMBERT Academic Publishing, 2011.
3. Коломеец Н. А., Павлов А. В. Свойства бент-функций, находящихся на минимальном расстоянии друг от друга // Прикладная дискретная математика. 2009. № 4. С. 5–20.

УДК 510.53

ОБ АЛГОРИТМИЧЕСКИХ И ТОПОЛОГИЧЕСКИХ СВОЙСТВАХ ОРБИТ КУСОЧНО-АФФИННЫХ ОТОБРАЖЕНИЙ

А. Н. Курганский

Рассматривается открытая проблема достижимости в одномерных кусочно-аффинных отображениях с двумя интервалами. Найдены частные случаи алгоритмической разрешимости рассматриваемой проблемы, сформулированные на языке топологических свойств орбит в таких системах.

Ключевые слова: кусочно-аффинные отображения, проблема достижимости.

При исследовании непрерывных динамических систем наряду с методами теории динамического хаоса, определяющими, например, такие свойства, как эргодичность, перемешивание, топологическая транзитивность и устойчивость, применение методов теории алгоритмов представляется также интересным и, главное, естественным и полезным с точки зрения приложения теории детерминированного хаоса в задачах криптографической защиты информации. Параллели между хаотическими и криптографическими системами мотивируют исследования в области применения дискретных аналогов непрерывных хаотических систем в задачах криптографического преобразования информации [1]. Однако математические связи относящихся к делу свойств дискретных систем и их непрерывных прототипов скорее находятся на уровне взаимосвязи узлов на память и запоминаемых фактов. В связи с этим на стыке теории динамических систем и теории алгоритмов представляет интерес фундаментальная проблема вычислительной универсальности непрерывных динамических систем и тесно с ней связанная проблема достижимости. Динамическая система является вычислительно универсальной, если существует интерпретация поведения системы, при которой моделируется машина Тьюринга. Проблему достижимости можно сформулировать так: существует ли для данной системы алгоритм, определяющий по данным точкам или областям фазового пространства, проходит ли через них одна и та же фазовая кривая.

Непрерывные хаотические динамические системы показывают сложное поведение, позволяющее предполагать для некоторых из них алгоритмическую неразрешимость проблемы достижимости из точки точки. До сих пор единственным способом доказательства алгоритмической неразрешимости является моделирование с помощью изучаемой системы универсальной машины Тьюринга. Публикации показывают в какой-то мере безуспешный поиск таких доказательств для различных непрерывных динамических систем. В связи с этим представляет интерес исследование вычислительной универсальности гибридных дискретно-непрерывных систем и систем с дискретным временем. Примеры таких исследований можно найти в [2]. На практике оказывается, что чем выше размерность фазового пространства, тем проще доказать универсальность системы. Поэтому представляют интерес системы низкой размерности. Например, в [3] для одномерных кусочно-элементарных отображений в базисе функций $\{x^2, x^3, x^{1/2}, x^{1/3}, 2x, x + 1, x - 1\}$, а также дробно-рациональных функций доказана алгоритмическая неразрешимость достижимости из точки точки. При этом уже долгое время остается открытой проблема достижимости в одномерных кусочно-аффинных отображениях [3] даже в простейшем случае двух интервалов. Настоящая работа посвящена кусочно-аффинным отображениям с двумя интервалами (2-РАМ'ам).

Не ясно, влечёт ли сопряжённость двух систем их эквивалентность с точки зрения проблемы достижимости. Во всяком случае, ответ на этот вопрос не очевиден. Поэтому, имея в виду как гипотезу алгоритмическую разрешимость проблемы достижимости в 2-РАМ, имеет смысл находить связи топологических и алгоритмических свойств орбит и проводить соответствующую классификацию систем. Например, если для динамической системы доказано свойство перемешивания или эргодичности, то проблема достижимости измеримой области (ненулевой меры) фазового пространства становится, очевидно, тривиальной. Пример менее тривиального утверждения приведён ниже.

Пусть X — отрезок $[0, 1]$ с отождествленными концами, т.е. $X = \mathbb{R}/\mathbb{Z}$, и отображение $f : X \rightarrow X$ таково, что $X = X_1 \cup X_2$ — разбиение на непересекающиеся интервалы и $f(x) = a_i x + b_i$, если $x \in X_i$; $a_i, b_i \in \mathbb{Q}$, $i = 1, 2$. Обозначим через $f^*(x) = \{f^n(x) : n \in \mathbb{N}\}$ орбиту точки x . Проблема достижимости из точки точ-

ки звучит так: существует ли алгоритм, определяющий по произвольным точкам $x_0, x_1 \in X$ принадлежность $x_1 \in f^*(x_0)$. Ограничимся рассмотрением только рациональных точек X по следующим соображениям. Кусочно-аффинное отображение $f(x) = 2x \pmod{1}$ является хаотическим для почти всех $x \in X$. Но ни одно число этого множества почти всех из X не представимо на компьютере, если под числом не понимать алгоритм, его порождающий. Если же под числом понимать произвольный порождающий его алгоритм, проблема достижимости для $f(x)$ оказывается тривиально алгоритмически неразрешимой в общем случае, несмотря на то, что отображение очень простое. Стоит при этом заметить, что некоторые сопряжённые $f(x)$ отображения показывают хаотическое поведение на множестве рациональных точек.

Теорема 1. Если f строго эргодическое, т. е. имеет единственную инвариантную меру, или, другими словами, плотности распределения орбит всех точек совпадают, то проблема достижимости алгоритмически разрешима.

Следствие 1. Если пространство инвариантных мер конечномерно, то проблема достижимости алгоритмически разрешима.

ЛИТЕРАТУРА

1. *Savchenko A. Ya., Kovalev A. M., Kozlovskii V. A., and Scherbak V. F.* Inverse dynamical systems in secure communication and its discrete analogs for information transfer // Proc. NDES 2003, May 18–22, Scuol/Schuls, Switzerland. P. 112–116.
2. *Asarin E., Mysore V., Pnueli A., and Schneider G.* Low dimensional hybrid systems — decidable, undecidable, don't know // Inform. Comput. 2012. V. 211. P. 138–159.
3. *Kurgansky O., Potapov I., and Sancho-Caparrini F.* Reachability problems in low-dimensional iterative maps // Int. J. Found. Comput. Sci. 2008. No. 19(4). P. 935–951.

УДК 519.7

О СТАТИСТИЧЕСКОЙ НЕЗАВИСИМОСТИ ПРОИЗВОЛЬНОЙ СУПЕРПОЗИЦИИ БУЛЕВЫХ ФУНКЦИЙ

О. Л. Мироненко

Доказаны достаточные условия статистической независимости суперпозиции произвольного числа булевых функций от подмножества своих аргументов.

Ключевые слова: суперпозиция булевых функций, статистическая независимость от подмножества аргументов.

Определение 1. Для любой булевой функции $f(x)$, где x — переменные со значениями в \mathbb{Z}_2^n , и для любого подмножества U её аргументов будем говорить, что $f(x)$ статистически не зависит от переменных множества U , если для любой её подфункции $f'(x')$, полученной фиксированием значений всех переменных в U , имеет место равенство $\Pr[f'(x') = 0] = \Pr[f(x) = 0]$.

В [1] доказано, что если булева функция $f_1(x)$ статистически не зависит от некоторого подмножества своих аргументов, то это свойство сохраняется для произвольной суперпозиции $g(f_1(x), y)$. Для суперпозиции $g(f_1(x), f_2(x), y)$ в [2] доказано более сложное достаточное условие сохранения статистической независимости: этим свойством, помимо f_1 и f_2 , должна обладать и их сумма $f_1 \oplus f_2$. В данной работе условие из [2] обобщается на случай произвольного числа внутренних функций суперпозиции.

Утверждение 1. Пусть x, y, z — переменные со значениями в \mathbb{Z}_2^n , \mathbb{Z}_2^m и \mathbb{Z}_2^k соответственно и для всех $1 \leq s \leq l$ и $1 \leq i_1 < i_2 < \dots < i_s \leq l$ функции $f_{i_1}(x, y) \oplus \dots \oplus \oplus f_{i_s}(x, y)$ статистически не зависят от переменных в x . Тогда для любой функции g от $l + k$ переменных суперпозиция $g(f_1(x, y), \dots, f_l(x, y), z)$ статистически не зависит от переменных в x .

Замечание 1. В ряде случаев (для конкретных функций g и ограничений на функции f_1, \dots, f_l) можно сократить количество достаточных условий утверждения; этот вопрос составляет предмет дальнейших исследований.

ЛИТЕРАТУРА

1. Колчева О. Л., Панкратова И. А. О статистической независимости суперпозиции булевых функций // Прикладная дискретная математика. Приложение. 2011. № 4. С. 11–12.
2. Колчева О. Л., Панкратова И. А. О статистической независимости суперпозиции булевых функций. II // Прикладная дискретная математика. Приложение. 2012. № 5. С. 14–15.

УДК 519.7

ВЕРХНЯЯ ОЦЕНКА АЛГЕБРАИЧЕСКОЙ ИММУННОСТИ НЕКОТОРЫХ БЕНТ-ФУНКЦИЙ ДИЛЛОНА

С. Ю. Филюзин

Получена верхняя оценка алгебраической иммунности некоторых бент-функций Диллона. Приводится степень бент-функций максимальной алгебраической иммунности, предложенных З. Ту и Й. Денгом.

Ключевые слова: булева функция, нелинейность, бент-функция, алгебраическая иммунность.

Путём детального анализа успешных способов взлома блочных и поточных шифров были определены свойства булевой функции, которыми она должна обладать для использования её в криптографических приложениях. На данный момент остаётся открытым вопрос о том, как совмещаются различные криптографические свойства у одной булевой функции. Данная работа посвящена изучению алгебраической иммунности при максимальной нелинейности булевой функции.

Булева функция от n переменных представима единственным образом в виде алгебраической нормальной формы (АНФ): $f(x_1, \dots, x_n) = \left(\bigoplus_{k=1}^n \bigoplus_{i_1, \dots, i_k} a_{i_1, \dots, i_k} x_{i_1} \cdot \dots \cdot x_{i_k} \right) \oplus a_0$, где $a_0, a_i \in \mathbb{Z}_2$. Степенью $\deg(f)$ булевой функции f называется число переменных в самом длинном слагаемом её АНФ. Алгебраической иммунностью $AI(f)$ булевой функции f называется минимальное целое число $d \geq 1$, такое, что существует булева функция g степени d , для которой выполняется равенство $fg = 0$ или $(f \oplus 1)g = 0$. Нелинейностью $nl(f)$ булевой функции f от n переменных называется расстояние Хэмминга от данной функции до множества аффинных функций от n переменных. Бент-функция — булева функция от n переменных (n чётно), обладающая максимальной нелинейностью равной $2^{n-1} - 2^{(n/2)-1}$.

На сегодняшний день полная классификация бент-функций не произведена, но предложены способы построения таких функций. Функцию вида $g : \text{GF}(2^k) \rightarrow \text{GF}(2)$ будем рассматривать как булеву, зафиксировав некоторый базис в поле $\text{GF}(2^k)$. В работе [1] Диллон приводит следующий способ построения бент-функций. Пусть

$g : \text{GF}(2^k) \rightarrow \text{GF}(2)$ — уравновешенная функция от k переменных и $g(0) = 0$. Тогда функция $f(x, y) = g(x \cdot y^{2^k-2})$ является бент-функцией от $2k$ переменных.

В данной работе исследуется алгебраическая иммунность некоторых бент-функций Диллона. В качестве уравновешенных рассматриваются линейные функции, как самые простые. Доказана

Теорема 1. Пусть функция $g : \text{GF}(2^k) \rightarrow \text{GF}(2)$ линейна, $g \neq \text{const}$ и f построена с помощью конструкции Диллона по функции g , а именно $f(x, y) = g(x \cdot y^{2^k-2})$. Тогда $\text{AI}(f) \leq \lceil k/2 \rceil + 1$. При $k = 2, \dots, 8$ оценка достигается.

Из теоремы заключаем, что алгебраическая иммунность бент-функций Диллона, построенных с помощью линейных булевых функций, отличается от максимально возможной почти в 2 раза.

В конструкции Диллона свойства бент-функции f зависят от выбора g . Возникает ряд вопросов — существуют ли такие g , чтобы $\text{AI}(f)$ была максимальной, а если существуют, то какими свойствами обладают.

В работе [2] предлагается способ построения таких g . Пусть $g : \text{GF}(2^k) \rightarrow \text{GF}(2)$, $\text{supp}(g) = \{\alpha^s, \dots, \alpha^{s+2^{k-1}-1}\}$, где α — примитивный элемент поля $\text{GF}(2^k)$ и $s \in \mathbb{N}$. Тогда бент-функция f , построенная с помощью конструкции Диллона, обладает максимальной алгебраической иммунностью. Доказано

Утверждение 1. Пусть $g : \text{GF}(2^k) \rightarrow \text{GF}(2)$, $\text{supp}(g) = \{\alpha^s, \dots, \alpha^{s+2^{k-1}-1}\}$, где α — примитивный элемент $\text{GF}(2^k)$ и $s \in \mathbb{N}$. Тогда для $k = 2, 3, \dots, 8$ верно $\deg(g) = k - 1$.

ЛИТЕРАТУРА

1. *Dillon J. F.* Elementary Hadamard difference sets. Ph. D. Thesis. Univ. of Maryland, 1974.
2. *Tu Z. and Deng Y.* A conjecture about binary strings and its applications on constructing Boolean functions with optimal algebraic immunity // *Designs, Codes and Cryptography*. 2011. V. 60. Iss. 1. P. 1–14.

УДК 519.6

ЭКВИВАЛЕНТНОСТЬ ПРИМИТИВНЫХ МНОЖЕСТВ

В. М. Фомичев

Исследована эквивалентность примитивных множеств натуральных чисел в связи с диофантовой проблемой Фробениуса. Эквивалентность используется для упрощения определения числа Фробениуса $g(a_1, \dots, a_k)$, а также всех чисел, не содержащихся в аддитивной полугруппе, порождённой множеством $\{a_1, \dots, a_k\}$.

Ключевые слова: функция Фробениуса, примитивное множество, аддитивная полугруппа чисел.

Основные обозначения:

\mathbb{N} — множество натуральных чисел, $k, n \in \mathbb{N}$, $A = \{a_1, \dots, a_k\} \subset \mathbb{N}$;

(a_1, \dots, a_k) — наибольший общий делитель чисел a_1, \dots, a_k ;

$\langle a_1, \dots, a_k \rangle$ — аддитивная полугруппа, порождённая множеством $\{a_1, \dots, a_k\}$;

$n\{a_1, \dots, a_k\} = \{na_1, \dots, na_k\}$;

$A^{(i)} = \{a_1, \dots, a_i\}$, $C(A^{(i)}) = d_i \mathbb{N}_0 \setminus \langle A^{(i)} \rangle$, $d_i = (a_1, \dots, a_i)$, $i = 2, \dots, k$.

Множество $A = \{a_1, \dots, a_k\}$ натуральных чисел, $k > 1$, называется примитивным, если $(a_1, \dots, a_k) = 1$.

Функция Фробениуса $g(a_1, \dots, a_k)$ определена на всех примитивных множествах $\{a_1, \dots, a_k\}$ как наибольшее натуральное число $t \notin \langle a_1, \dots, a_k \rangle$ — число, не представимое в виде линейной комбинации чисел a_1, \dots, a_k (рассматриваются линейные комбинации только с целыми неотрицательными коэффициентами):

$$g(a_1, \dots, a_k) = \max\{t \in \mathbb{N} : t \neq n_1 a_1 + \dots + n_k a_k\}.$$

Число Фробениуса $g(A) = \max C(A)$, где $C(A) = C(A^{(k)})$ — множество всех натуральных чисел, не представимых линейными комбинациями чисел a_1, \dots, a_k . Задача определения $g(a_1, \dots, a_k)$ известна как диофантова проблема Фробениуса (ПФ). Задача определения множества $C(A)$ называется расширенной проблемой Фробениуса (РПФ).

При $k = 2$ описание множества $C(A)$ и формула числа Фробениуса даны Дж. Сильвестром [1] ещё в 1884 г., формула числа Фробениуса имеет вид $g(a_1, a_2) = a_1 a_2 - a_1 - a_2$.

При $k > 2$ алгоритмы РПФ до недавнего времени не исследовались, в то же время активно изучался порядок множества $C(A)$ и некоторые его свойства [2, гл. 5]. В [3] представлен алгоритм РПФ и дана оценка его сложности.

Намного активнее исследовалась ПФ. Вместе с тем прогресс в получении формул следует признать умеренным, получены формулы только для множеств частного вида при $k = 3, 4$. Эти трудности в определённой мере объяснимы; показано [4], что уже при $k = 3$ не существует конечного числа полиномов, позволяющих выразить через них число Фробениуса $g(a_1, a_2, a_3)$ с помощью разбиения области определения.

Более продуктивным стал алгоритмический подход к определению $g(a_1, \dots, a_k)$. Построен ряд алгоритмов как при $k = 3$ (O. Rodseth, J. Davison и др. [2]), так и при произвольном k . В работах [5, 6] разработан алгоритм вычисления функции Фробениуса при любом k с помощью экспоненцирования неотрицательной матрицы M порядка a_k и определения её экспонента с использованием соотношения $g(a_1, \dots, a_k) = \exp M - a_k$. Вычислительная сложность алгоритма в битовых операциях равна $O(a_k^3)$, требуемая память — $O(a_k^2 \log a_k)$ битов. В [7] предложено (без обоснования корректности) улучшение этого алгоритма со сложностью $O(a_k^2)$.

В [8] построен теоретико-графовый алгоритм определения $g(a_1, \dots, a_k)$ со сложностью $O(a_1(k + \log a_1))$ арифметических операций. В [9] эта оценка снижена до величины порядка $O(ka_1)$ операций.

В [3] на основе исследования отношений эквивалентности предложена редукция РПФ и ПФ для множества A к РПФ и ПФ соответственно для собственного подмножества порядка h , где $2 \leq h \leq \min(k, a_1)$. Редукция позволила снизить оценку $O(ka_1)$ до величины $O(l(A) + ha_1)$ арифметических операций, $l(A) \leq k$ — характеристика множества A .

Исследования отношений эквивалентности в данной работе получили дальнейшее развитие.

1. Эквивалентность множеств, $k > 2$

Примитивное множество $A = \{a_1, \dots, a_k\}$ рассмотрим как возрастающую последовательность, где $1 < a_1 < \dots < a_k$. Числа a_i и a_j множества A назовём a_1 -эквивалентными, если $a_i = a_j \pmod{a_1}$. Примитивное множество $A = \{a_1, \dots, a_k\}$ разбивается на h классов a_1 -эквивалентности, где $2 \leq h \leq \min\{k, a_1\}$. Трансверсал множества A по отношению a_1 -эквивалентности, состоящий из наименьших чисел классов, назовём a_1 -трансверсалом множества A и обозначим A/a_1 . Обозначим $W(A/a_1)$ последовательность порядковых номеров $i \geq 2$ чисел a_1 -трансверсала.

Множество A называется приведённым, если $a_i \notin \langle A^{(i-1)} \rangle$, $i = 2, \dots, k$. Неприведённое примитивное множество A содержит приведённые примитивные подмножества. Построим приведённое подмножество $B_A \subseteq A$, называемое A -базисом: $a_1 \in B_A$; $a_2 \in B_A$, если и только если a_2 не кратно a_1 ; пусть из $\{a_1, \dots, a_{i-1}\}$ в подмножество B_A включены числа b_1, \dots, b_j , тогда $a_i \in B_A$, если и только если $d_{i-1} > d_i$ или $(d_{i-1} = d_i$ и $a_i \in C(b_1, \dots, b_j))$, $i = 3, \dots, k$. Обозначим $W(B_A)$ последовательность порядковых номеров i чисел A -базиса.

Наименьшее натуральное $p(A)$, такое, что $d_{p(A)} = 1$, называется индексом примитивности множества A . Положим $p(A) = p$, тогда $p \leq k$, $d_p = \dots = d_k = 1$. Последовательность порядковых номеров $i \in \{2, \dots, p\}$, таких, что $d_{i-1} > d_i$, обозначим $L(A)$.

Подмножества D и D' множества A эквивалентны (g -эквивалентны), если $\langle D \rangle = \langle D' \rangle$ (если $g(D) = g(D')$); будем это отношение обозначать $D \cong D'$ (соответственно $D \cong^g D'$). Тогда $D \cong D'$, если и только если $C(D) = C(D')$, и если $D \cong D'$, то $g(D) = g(D')$.

Утверждение 1. $B_A \subseteq A/a_1 \subseteq A$, $B_A \cong A/a_1 \cong A$, где $|B_A| \leq |A/a_1| \leq \min\{k, a_1\}$; оценка достижима.

Следствие 1. Множество $A = \{a_1, \dots, a_k\}$ содержит эквивалентное подмножество порядка не больше $\min\{k, a_1\}$.

Пример 1 (построение A/a_1). Для $A = \{5, 13, 18, 20, 38\}$ считаем по mod 5: $5 \equiv 20$, $13 \equiv 18 \equiv 38$, тогда $A/5 = \{5, 13\}$.

Пусть $a_1 \geq 3$ и $A = A/a_1$, т.е. $k \leq a_1$ и числа a_1, \dots, a_k несравнимы по модулю a_1 ; $L(A) = \{n_2, \dots, n_r\}$, $1 < r < k$, т.е. $d_{n_j-1} > d_{n_j}$, $j = 2, \dots, r$. Тогда A разбивается на r отрезков A_1, \dots, A_r , где $A_j = (a_{n_j}, \dots, a_{n_{j+1}-1})$, $j = 1, \dots, r$, $n_1 = 1$, $n_r = p$, $n_{r+1} = k + 1$.

Обозначим $J_1 = A_1 \setminus \{a_1\}$, $J(A)$ — множество чисел $a_i \in A$, таких, что $d_{i-1} = d_i$ и a_i не делит ни одно число из $C(A^{(i-1)})$ (т.е. $J(A) \subseteq \{n_2 + 1, \dots, k\} \setminus \{n_2, \dots, n_r\}$), $W(J)$ — последовательность порядковых номеров чисел из $J(A)$.

Далее без ущерба для общности считаем, что $n_2 = 2$. Приведём некоторые известные свойства.

Утверждение 2.

- 1) Если $B \subset A$ и $A \setminus B \subset \langle B \rangle$, то $A \cong B$.
- 2) Множество $A^{(i)}/d_i = \{a_1/d_i, \dots, a_i/d_i\}$ примитивное, $\langle A^{(i)} \rangle = d_i \langle A^{(i)}/d_i \rangle$, $i = 2, \dots, k$.
- 3) $g(a_1, \dots, a_k) \leq (a_1 - 1)(a_k - 1) - 1$ [2, теорема 3.1.1].

Теорема 1. $A \cong A \setminus J(A)$; если $n_2 > 2$, то $A \cong A \setminus (J_1 \cup J(A))$.

Обозначим $\gamma_j = d_{n_j}((a_1/d_{n_j} - 1)(a_{n_j}/d_{n_j} - 1) - 1)$, $l_1 = 1$, l_j — наибольший номер, такой, что $n_j \leq l_j < n_{j+1}$ и $a_{l_j} \leq \gamma_j$ (при $a_1 \geq 3$ условия корректны в силу утверждения 2; $S_1 = (a_1)$, $S_j = (a_{n_j}, \dots, a_{l_j})$, т.е. отрезок S_j составлен из первых (наименьших) $l_j - n_j + 1$ чисел отрезка A_j , $j = 2, \dots, r$; $S(A)$ есть конкатенация отрезков S_1, S_2, \dots, S_r , $l(A) = |S(A)|$.

Следствие 2. $A \cong S(A)$, $l(A) \leq \min\{k, a_1 a_p - a_1 - 2a_p + p\}$, где $p = p(A)$.

Алгоритмы построения A/a_1 и $A \setminus S(A)$ являются полиномиальными. Алгоритм построения $A \setminus J(A)$ экспоненциальный, по сложности он равносильен решению расширенной проблемы Фробениуса [3].

Пример 2. Пусть $A = \{5, 13, 18, 20, 57\}$. Вычисляем $A/5 = \{5, 13, 57\}$, $J(A/5) : L(A/5) = \{13\}$, $r = 2$, $A_1 = \{5\}$, $A_2 = \{13, 52\}$, $\gamma_2 = 47 < 57$. Тогда $J(A/5) = \{57\}$, $(A/5) \setminus J(A/5) = \{5, 13\}$. Окончательно $A \cong \{5, 13\}$.

2. Эквивалентность примитивных пар чисел

Обозначим через $[A]$ класс g -эквивалентности, содержащий примитивное множество A ; $[A]_m$ — класс всех примитивных множеств порядка m , g -эквивалентных A , $m = 2, 3, \dots$

Теорема 2. Пусть (a, b) — примитивная пара чисел. Тогда

$$[(a, b)]_2 = (a + \delta, b + \varepsilon) : a - b + 1 < \varepsilon < 0, \delta = (1 - a)\varepsilon / (b + \varepsilon - 1) \in \mathbb{N}; \\ 1 - a < \delta < 0, \varepsilon = (1 - b)\delta / (a + \delta - 1) \in \mathbb{N}.$$

Следствие 3.

Если $a > 2$, то $(2, g(a, b) + 2) \in [(a, b)]_2$; если $a > 3$, то $(3, (g(a, b) + 3)/2) \in [(a, b)]_2$.

Пример 3. Построим множество $[(13, 22)]_2$, $g(13, 22) = 251$. При $\delta > 0$ имеем $-8 < \varepsilon < 0$, при $\varepsilon > 0$ имеем $-12 < \delta < 0$.

ε	-7	-6	-5	-4	-3	-2	-1
δ	6	72/15	60/16	48/17	2	24/19	12/20

ε	21/11	42/10	7	84/8	15	21	147/5	42	63	105	231
δ	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11

Согласно таблице, $(\delta, \varepsilon) \in \{(2, -3), (6, -7)\}$. Пары $(13, 22)$ g -эквивалентна примитивная приведённая упорядоченная пара $(15, 19)$. Пара $(19, 15)$ исключена из класса $[(13, 22)]_2$, так как не упорядочена. Тогда $(\delta, \varepsilon) \in \{(-3, 7), (-5, 15), (-6, 21), (-8, 42), (-9, 63), (-10, 105), (-11, 231)\}$ и $(13, 22)$ g -эквивалентна парам $(10, 29)$, $(8, 37)$, $(7, 43)$, $(5, 64)$, $(4, 85)$, $(3, 127)$ и $(2, 253)$. Класс g -эквивалентности $[(13, 22)]_2$ содержит (вместе с $(13, 22)$) 9 пар.

Вывод: Отношения эквивалентности примитивных множеств позволяют сократить сложность решения РПФ и ПФ с помощью полиномиального алгоритма редукции примитивного множества A к подмножеству $B = (A/a_1) \setminus S(A/a_1)$. Сложность экспоненцирования неотрицательной матрицы $[5, 6]$ имеет в сочетании с редукцией оценку $O(a^3)$, где $a = \max B$. Оценка Боккера — Липтак [9] понижается до величины порядка $O(l(A) + ha_1)$, где $h = |B|$.

ЛИТЕРАТУРА

1. *Sylvester J. J.* Problem 7382 // *Mathematical Questions from the Educational Times*. 1884. V. 37. P. 26.
2. *Alfonso J. R.* *The Diophantine Frobenius problem*. Oxford University Press, 2005.
3. *Фомичев В. М.* Решение диофантовой проблемы Фробениуса // *Дискретная математика*. 2013. № 2.
4. *Curtis F.* On formulas for the Frobenius number of a numerical semigroup // *Math. Scand.* 1990. V. 67. P. 190–192.
5. *Heap B. R. and Lynn M. S.* A graph-theoretic algorithm for the solution of a linear diophantine problem of Frobenius // *Numerische Math.* 1964. No. 6. P. 346–354.
6. *Heap B. R. and Lynn M. S.* On a linear diophantine problem of Frobenius: an improved algorithm. // *Numerische Math.* 1965. No. 7. P. 226–231.

7. Bogart C. Calculating Frobenius numbers with Boolean Toeplitz matrix multiplication // For Dr. Cull, CS 523, March 17, 2009. Oregon State University.
8. Nijenhuis M. A minimal-path algorithm for the “money changing problem” // The American Mathematical Monthly. 1979. V. 86. P. 832–835.
9. Bocker S. and Liptak Z. The “money changing problem” revisited: computing the Frobenius number in time $O(ka_1)$. Technical Report No. 2004-2, Univ. of Bielefeld, Technical Faculty, 2004.

УДК 519.7

ИТЕРАТИВНАЯ КОНСТРУКЦИЯ APN-ФУНКЦИЙ¹

А. А. Фролова

Векторные булевы функции F и G назовём γ -эквивалентными, если для каждой пары векторов $a \neq 0$, b уравнения $F(x) \oplus F(x \oplus a) = b$ и $G(x) \oplus G(x \oplus a) = b$ одновременно имеют или не имеют решений. Установлено, что все классы γ -эквивалентности APN-функций от n переменных имеют мощность 2^{2n} . Предложена итеративная конструкция APN-функций.

Ключевые слова: векторная булева функция, APN-функция, γ -эквивалентность, итеративная конструкция.

Булевой функцией от n переменных называется любое отображение $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$. Весом Хэмминга $\text{wt}(f)$ булевой функции f называется количество единиц в векторе её значений. Векторной булевой функцией F называется любое отображение $F : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$. Векторную функцию можно рассматривать как набор из m координатных булевых функций от n переменных, т. е. $F = (f_1, \dots, f_m)$.

Булевы функции, используемые в криптографических приложениях, должны обладать рядом специальных свойств для обеспечения стойкости к некоторым видам криптоанализа. В работе [1] определено следующее требование к функции. Векторная функция F называется δ -дифференциально равномерной, если для любых векторов $a \neq 0$, b уравнение $F(x) \oplus F(x \oplus a) = b$ имеет не более δ решений. Для обеспечения стойкости шифра к дифференциальному криптоанализу необходимо использовать δ -дифференциально равномерные векторные булевы функции с малым значением δ .

Далее рассматриваем только случай $n = m$. В этом случае минимальное возможное δ равно двум. APN-функцией (Almost Perfect Nonlinear) называется 2-дифференциально равномерная векторная функция. В работе [2] приведён обзор по известным APN-функциям. Открытыми вопросами остаются оценки количества и новые способы построения APN-функций.

Пусть $F : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^n$. Для F и любого вектора $a \neq 0$ определяется множество

$$B_a(F) = \{F(x) \oplus F(x \oplus a) : x \in \mathbb{Z}_2^n\}.$$

Для F строится булева функция γ_F от $2n$ переменных следующим образом:

$$\gamma_F(a, b) = \begin{cases} 1, & \text{если } a \neq 0 \text{ и } b \in B_a(F), \\ 0 & \text{иначе.} \end{cases}$$

Известно, что F — APN-функция тогда и только тогда, когда $\text{wt}(\gamma_F) = 2^{2n-1} - 2^{n-1}$. Пусть $F, F' : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^n$.

¹Работа поддержана грантом РФФИ, проект № 12-01-31097.

Определение 1. Функции F и F' назовём γ -эквивалентными, если $\gamma_F = \gamma_{F'}$.

Нетрудно убедиться, что γ -эквивалентность является отношением эквивалентности на множестве всех векторных булевых функций. Следовательно, множество функций распадается на непересекающиеся классы.

Получены следующие результаты.

Теорема 1. Пусть F — APN-функция от n переменных. Тогда все функции $F_{c,d}(x) = F(x \oplus c) \oplus d$, где $c, d \in \mathbb{Z}_2^n$, являются APN-функциями, γ -эквивалентными F . Кроме того, все функции $F_{c,d}$ попарно различны.

Теорема 2. Пусть γ — булева функция от $2n$ переменных, $\gamma = \gamma_F$ для некоторой APN-функции F от n переменных. Тогда существует не более 2^{2n} APN-функций с такой γ .

Следствие 1. В каждом классе γ -эквивалентности APN-функций от n переменных ровно 2^{2n} различных функций.

Опишем итеративную конструкцию APN-функции от $n + 1$ переменных из двух APN-функций и двух булевых функций от n переменных.

Теорема 3. Пусть F и G — APN-функции от n переменных, f и g — булевы функции от n переменных. Пусть S — векторная булева функция от $n + 1$ переменных, определённая как

$$S(x, x_{n+1}) = ((x_{n+1} \oplus 1)F(x) \oplus x_{n+1}G(x), (x_{n+1} \oplus 1)f(x) \oplus x_{n+1}g(x)),$$

где $x \in \mathbb{Z}_2^n$, $x_{n+1} \in \mathbb{Z}_2$. Тогда S — APN-функция, если выполнено условие

$$\begin{aligned} \text{для всех } x, y, a \in \mathbb{Z}_2^n, a \neq 0, \text{ таких, что } F(x) \oplus F(x \oplus a) = G(y) \oplus G(y \oplus a), \\ \text{выполняется } f(x) \oplus f(x \oplus a) \neq g(y) \oplus g(y \oplus a). \end{aligned} \quad (1)$$

Следствие 2. Пусть F и G — APN-функции от n переменных, f и g — булевы функции от n переменных, удовлетворяющие условию (1). Тогда функции $F'(x) = F(x \oplus c') \oplus d'$, $G'(x) = G(x \oplus c'') \oplus d''$, $f' = f(x \oplus c') \oplus d_1$, $g'(x) = g(x \oplus c'') \oplus d_2$ удовлетворяют условию (1) для любых $c', c'', d', d'' \in \mathbb{Z}_2^n$, $d_1, d_2 \in \mathbb{Z}_2$.

Открытым остаётся вопрос, как выбрать APN-функции F , G и булевы функции f , g , которые бы удовлетворяли условию (1). При малых n экспериментально показано, что для любой APN-функции F найдётся достаточно много функций G , f , g , удовлетворяющих (1). Можно сформулировать гипотезу.

Гипотеза 1. Для любой APN-функции F от n переменных найдутся APN-функция G и булевы функции f , g от n переменных, удовлетворяющие условию (1).

ЛИТЕРАТУРА

1. Nyberg K. Differentially uniform mappings for cryptography // Eurocrypt 1993. LNCS. 1994. V. 765. P. 55–64.
2. Тужилин М. Э. Почти совершенные нелинейные функции // Прикладная дискретная математика. 2009. № 3. С. 14–20.

УДК 519.719.325

К ОПРЕДЕЛЕНИЮ СТЕПЕНИ НЕЛИНЕЙНОСТИ ДИСКРЕТНОЙ ФУНКЦИИ НА ЦИКЛИЧЕСКОЙ ГРУППЕ¹

А. В. Черемушкин

Предлагается аддитивный подход к определению степени нелинейности дискретных функций, заданных на циклической группе. Показано, что степень нелинейности конечна, если и только если порядок группы есть степень простого числа; найдены верхние оценки степени нелинейности. Показано, что для полиномиальных функций над кольцом \mathbb{Z}_{p^n} степень нелинейности функции совпадает с минимальной степенью многочлена, задающего эту функцию.

Ключевые слова: дискретные функции, степень нелинейности.

В работе [1] предложен аддитивный подход к определению степени нелинейности функции на основе свойств конечных производных. Его суть заключается в следующем. Для функций $F : G \rightarrow H$, у которых на множествах G и H заданы структуры абелевых групп, производная по направлению $\Delta_a F$, $a \in G$, функции F определяется равенствами

$$\Delta_a F(x) = F(x + a) - F(x),$$

где $x \in G$. Степенью нелинейности функции $F : G \rightarrow H$ (обозначается $\text{dl } F$) называется минимальное натуральное число m , такое, что

$$\Delta_{a_1} \dots \Delta_{a_{m+1}} F(x) = 0$$

при всех $a_1, \dots, a_{m+1}, x \in G$. Если такого числа m не существует, то полагаем $\text{dl } F = \infty$.

В случае элементарных абелевых p -групп степень нелинейности функции p^n -значной логики полностью определяется свойствами операции сложения. Поэтому при любом способе задания на этой группе операции умножения так, чтобы в результате получилось поле из p^n элементов, степень нелинейности функций инвариантна по отношению к выбору операции умножения.

В случае произвольных абелевых групп вопрос о свойствах параметра $\text{dl } F$ остается открытым. В данной работе изучается случай циклических групп.

Показано, что параметр $\text{dl } F$ в случае, когда на множестве аргументов и значений заданы структуры циклических групп, может принимать конечные значения, только когда порядки групп являются примарными числами.

Теорема 1. Если $F : G^m \rightarrow H$, где G и H — циклические группы порядков $g \geq 2$ и $h \geq 2$ соответственно, причем $\text{dl } F < \infty$, то при некотором простом $p \geq 2$ выполнены равенства $g = p^n$ и $h = p^k$ при некоторых $n \geq 1$ и $k \geq 1$.

Следующая теорема показывает, что введенное выше «аддитивное» определение степени нелинейности корректно и параметр $\text{dl } F$ принимает конечное значение для любой функции, заданной на циклических группах примарного порядка.

Теорема 2. Если $F : G^m \rightarrow H^t$, $G = \mathbb{Z}_{p^n}$, $H = \mathbb{Z}_{p^k}$, $p > 2$, $n \geq 1$, $k \geq 1$, $m \geq 1$, $t \geq 1$, и $F = (F_1, \dots, F_t)$, где $F_i : G^m \rightarrow H$, $1 \leq i \leq t$, — соответствующие координатные функции, то

$$\text{dl } F = \max\{\text{dl } F_i : 1 \leq i \leq t\} \leq m(p^n - (k - 1)(p - 1)p^{n-1} - 1).$$

¹Работа выполнена при поддержке гранта Президента РФ НШ № 6260.2012.10.

Преимущество данного подхода к определению степени нелинейности заключается в том, что он полностью определяется свойствами только операции сложения. Вместе с тем в случае циклических групп, в отличие от элементарных абелевых групп, вопрос о способе выбора операции умножения представляется не таким однозначным. Если естественным образом рассматривать циклические группы примарного порядка как аддитивные группы колец вычетов с имеющимися в них операциями умножения, то определение степени нелинейности через степень многочлена является неудобным, так как не всякая функция F может быть задана многочленом (или набором многочленов координатных функций). Полиномиальные функции над кольцом вычетов, то есть функции, которые могут быть заданы многочленом над этим кольцом, составляют относительно малую долю функций [2, 3]. Следует отметить, что для полиномиальных функций над кольцом \mathbb{Z}_{p^n} степень нелинейности функции совпадает с минимальной степенью многочлена, задающего эту функцию.

Теорема 3. Если $F : G^m \rightarrow G$ — полиномиальная функция над кольцом $G = \mathbb{Z}_{p^n}$, $p > 2$, $n \geq 1$, $m \geq 1$, и $P(x_1, \dots, x_n) \in \mathbb{Z}_{p^n}[x_1, \dots, x_n]$ — многочлен минимальной степени, задающий эту функцию, то степень нелинейности совпадает со степенью многочлена $P(x_1, \dots, x_n)$:

$$\text{dl } F = \text{deg } P.$$

Из определения степени нелинейности с очевидностью вытекает

Теорема 4. Если G и H — циклические p -группы примарного порядка и R — подгруппа в G , то для степеней нелинейности функции $F : G \rightarrow H$ и её ограничения на подгруппу R , $F|_R : R \rightarrow H$, выполнено неравенство $\text{dl } (F|_R) \leq \text{dl } F$.

Подробное изложение представленных результатов можно найти в [4].

ЛИТЕРАТУРА

1. Черемушкин А. В. Аддитивный подход к определению степени нелинейности дискретной функции // Прикладная дискретная математика. 2010. № 2(8). С. 22–33.
2. Keller G. and Olson F. Counting polynomial functions (mod p^n) // Duke Math. J. 1968. V. 35. P. 835–838.
3. Chen Z. On polynomial functions from $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \dots \times \mathbb{Z}_{n_r}$ to \mathbb{Z}_m // Discrete Math. 1996. V. 162. P. 67–76.
4. Черемушкин А. В. Аддитивный подход к определению степени нелинейности дискретной функции на циклической группе примарного порядка // Прикладная дискретная математика. 2013. № 2(20). С. 26–38.

УДК 621.391: 519.728

ЭКОНОМНОЕ ПРЕДСТАВЛЕНИЕ НЕДООПРЕДЕЛЁННЫХ ДАННЫХ И ДИЗЬЮНКТИВНЫЕ КОДЫ¹

Л. А. Шоломов

Предложены экономные представления недоопределённых данных, позволяющие полностью восстанавливать исходные данные. Установлена их связь с дизьюнктивными кодами, получены оценки длины представлений.

Ключевые слова: представление недоопределённых данных, дизьюнктивный код, свободная от покрытий матрица.

¹Работа поддержана ОНИТ РАН по программе фундаментальных исследований.

Задан алфавит $A_0 = \{a_0, a_1, \dots, a_{m-1}\}$ основных символов. Каждому непустому $T \subseteq M = \{0, 1, \dots, m-1\}$ поставлен в соответствие *недоопределённый символ* a_T . Его *доопределением* считается всякий основной символ a_i , $i \in T$. Символ a_M , доопределимый любым a_i , называется *неопределённым* и обозначается $*$. Выделена система \mathcal{T} некоторых непустых подмножеств T множества M и с ней связан *недоопределённый алфавит* $A = A_{\mathcal{T}} = \{a_T : T \in \mathcal{T}\}$. Подробнее о недоопределённых данных см. в [1].

Задавшись натуральным числом s , припишем каждому $a_i \in A_0$ набор $\lambda_i = (\lambda_i(1), \dots, \lambda_i(s)) \in \{0, 1\}^s$ — код символа a_i , а каждому $a_T \in A$ — набор $\lambda_T = (\lambda_T(1), \dots, \lambda_T(s)) \in \{0, 1, *\}^s$. Обозначим через Λ и $\tilde{\Lambda}$ матрицы со столбцами λ_i , $i \in M$, и λ_T , $T \in \mathcal{T}$, соответственно. Скажем, что пара $(\Lambda, \tilde{\Lambda})$ задаёт для алфавита A *двоичное представление* (размерности s), если при всех i и T столбец λ_i доопределяет λ_T тогда и только тогда, когда $i \in T$. В случае, когда $\tilde{\Lambda}$ — матрица в двухбуквенном алфавите $\{0, *\}$, представление будем называть *строго двоичным*.

Будем говорить, что *множество столбцов* T матрицы Λ (i) *покрывает*, (ii) *инверсно покрывает*, (iii) *дважды покрывает* столбец λ_j , если (i) дизъюнкция столбцов λ_i , $i \in T$, покрывает λ_j , (ii) дизъюнкция инверсий столбцов λ_i , $i \in T$, покрывает инверсию столбца λ_j , (iii) множество столбцов T покрывает и инверсно покрывает λ_j . Матрица Λ *свободна от \mathcal{T} -покрытий* (*двойных \mathcal{T} -покрытий*), если для любого $T \in \mathcal{T}$ множество столбцов T не покрывает (не покрывает дважды) ни одного λ_j , $j \notin T$.

Теорема 1. Двоичное (строго двоичное) представление $(\Lambda, \tilde{\Lambda})$ с матрицей кодирования Λ существует для алфавита $A = A_{\mathcal{T}}$ тогда и только тогда, когда она свободна от двойных \mathcal{T} -покрытий (\mathcal{T} -покрытий).

По матрице Λ из теоремы можно эффективно (полиномиально) строить (строгие) представления недоопределённых последовательностей и восстанавливать по ним исходные последовательности, не используя $\tilde{\Lambda}$.

Скажем, что система \mathcal{Z} подмножеств множества M образует *конъюнктивный базис* (*обобщённый конъюнктивный базис*) системы \mathcal{T} , если каждое множество $T \in \mathcal{T}$ может быть получено как пересечение некоторых множеств (множеств либо их дополнений) из \mathcal{Z} . Строке v , $v = 1, \dots, s$, матрицы Λ сопоставим множество $Z_v \subseteq M$, образованное номерами единичных разрядов строки. Положим $\mathcal{Z} = \{Z_1, \dots, Z_s\}$, $\mathcal{Z}' = \{\bar{Z}_1, \dots, \bar{Z}_s\}$, где чёрточка означает дополнение. Для построения матриц, свободных от покрытий (либо двойных покрытий), может быть использован следующий факт.

Теорема 2. Матрица Λ свободна от \mathcal{T} -покрытий (двойных \mathcal{T} -покрытий) тогда и только тогда, когда соответствующая ей система \mathcal{Z}' (система \mathcal{Z}) образует конъюнктивный базис (обобщённый конъюнктивный базис) системы \mathcal{T} .

Пусть $s(A)$ и $s_0(A)$ означают наименьшую размерность соответственно двоичных и строго двоичных представлений алфавита A . Недоопределённые данные, с которыми имеют дело в приложениях, обычно помимо неопределённого символа $*$ используют лишь символы, имеющие небольшое число доопределений. Обозначим через $s(m, n, t)$ максимальную из размерностей $s(A)$ алфавитов A , для которых $|A_0| = m$, $|A| = n$ и каждый символ $a_T \in A$ имеет не более t доопределений либо является неопределённым. Аналогичную величину при использовании $s_0(A)$ обозначим $s_0(m, n, t)$.

Теорема 3. Справедливы оценки

$$s(m, n, t) \leq s_0(m, n, t) \leq e(t+1) \ln(mn) + 1.$$

Используя очевидное соотношение $n \leq m^t$, получаем границу вида $O(t^2 \ln m)$. При малых t эта величина существенно меньше длины m естественного задания недоопределённых символов a_T посредством характеристического набора множества T .

Теорема 4. При выполнении условия $t = o(\log n / \log \log n)$ имеют место оценки

$$s(m, n, t) \gtrsim \frac{(t-1) \log n}{2(2 \log(t-1) + c)}, \quad s_0(m, n, t) \gtrsim \frac{(t+1) \log n}{2(2 \log t + c)},$$

где $\log x = \log_2 x$, $c = \log(3e/4) < 1,027$.

При естественном условии $m \leq n$ верхние и нижние оценки теорем 3 и 4 различаются по порядку в $\log t$ раз.

В случае, когда система \mathcal{T} состоит из всех t -элементных подмножеств множества M , \mathcal{T} -дизъюнктивную матрицу называют *t-дизъюнктивной*. Множество её столбцов образует *t-дизъюнктивный код*. Дизъюнктивные (superimposed) коды, введённые в [2], находят широкое применение в информатике. Эффективные методы построения t -дизъюнктивных кодов, развитые в [2, 3] и других работах, могут быть использованы для эффективного представления недоопределённых данных.

Более подробное изложение представленных результатов можно найти в [4].

ЛИТЕРАТУРА

1. Шоломов Л. А. Элементы теории недоопределённой информации // Прикладная дискретная математика. Приложение. 2009. № 2. С. 18–42.
2. Kautz W. H. and Singleton R. C. Nonrandom binary superimposed codes // IEEE Trans. Inform. Theory. 1964. V. 10. No. 4. P. 363–377.
3. Kumar R., Rajagopalan S., and Sahai A. Coding construction for blacklisting problems without computational assumptions // CRYPTO-99. LNCS. 1999. V. 1666. P. 609–623.
4. Шоломов Л. А. Двоичные представления недоопределённых данных и дизъюнктивные коды // Прикладная дискретная математика. 2013. № 1(19). С. 17–33.

Секция 2

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 519.7

О ПРЕДСТАВЛЕНИИ S-БЛОКОВ
ПРИ РЕАЛИЗАЦИИ В БЛОЧНЫХ ШИФРАХ

В. А. Виткуп

Рассматривается предложенный недавно способ разбиения S-блоков для защиты от атак по сторонним каналам. Известно, что для всех классов эквивалентности S-блоков, кроме одного, такое разбиение возможно. Доказано, что для этого одного класса не существует искомого разбиения.

Ключевые слова: S-блок, векторные булевы функции, аффинная эквивалентность.

Многие криптографические алгоритмы уязвимы к атакам по сторонним каналам, направленным на слабости в практической реализации алгоритма. В качестве мер противодействия используются методы, маскирующие входные данные так, чтобы вычисления не зависели от них в явном виде.

В [1] предложен следующий способ маскирующей реализации S-блока. Рассмотрим S-блок $n \times n$. Пусть $x = (x_1, \dots, x_n)$, где $x_i \in \mathbb{Z}_2$. Рассмотрим векторную функцию $S : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^n$, $S = (f_1, \dots, f_n)$, где f_1, \dots, f_n — булевы функции от n переменных. Разбиваем переменные x_i каждую на r булевых переменных: $x_i = \sum_{j=1}^r x_{ij}$. Пусть $v = (x_{11}, \dots, x_{nr})$. Разбиваем функцию S на r векторных функций $S_i : \mathbb{Z}_2^{nr} \rightarrow \mathbb{Z}_2^n$ так, чтобы выполнялось $S(x) = \sum_i S_i(v)$. Такое разбиение векторной функции S обозначим $P(S)$.

Введём следующие условия для разбиения.

1. *Неполнота*: блок S_i не должен зависеть от переменных x_{ki} , $k = 1, \dots, n$.
2. *Взаимная однозначность*: функция $S^* : \mathbb{Z}_2^{nr} \rightarrow \mathbb{Z}_2^{nr}$, $S^* = (S_1, \dots, S_r)$, является взаимно однозначной.

Разбиение $P(S)$, удовлетворяющее этим двум условиям, называется *допустимым*.

Две векторных функции S и \bar{S} называются *аффинно эквивалентными*, если существует пара невырожденных аффинных преобразований A и B , таких, что $S = B \circ \bar{S} \circ A$. Отношение аффинной эквивалентности разбивает множество всех взаимно однозначных S-блоков на непересекающиеся классы. Множество S-блоков 3×3 содержит 4 класса, $\mathcal{A}_1^3, \mathcal{Q}_1^3, \mathcal{Q}_2^3, \mathcal{Q}_3^3$. В таблице приведены их представители.

Класс	Представитель
\mathcal{A}_1^3	(x, y, z)
\mathcal{Q}_1^3	$(x, y, xy + z)$
\mathcal{Q}_2^3	$(x, y + xz, z + xy + xz)$
\mathcal{Q}_3^3	$(xy + xz + yz, x + y + xy + yz, x + z + yz)$

Теорема 1 [1]. Если для некоторой векторной функции существует допустимое разбиение, то для любой аффинно эквивалентной ей функции также существует допустимое разбиение.

Построить разбиение и добиться выполнения условия неполноты нетрудно; сложность представляет свойство взаимной однозначности, которое требует отдельной проверки для каждого полученного разбиения. Для классов \mathcal{A}_1^3 , \mathcal{Q}_1^3 и \mathcal{Q}_2^3 допустимые разбиения найдены в работе [1]. Чтобы достигнуть взаимной однозначности, в функции из разбиения S-блока добавляются пары так называемых корректирующих слагаемых, комбинацией которых можно получить всевозможные разбиения $P(S)$, удовлетворяющие условию 1. Для S-блоков 3×3 существует всего 54 таких слагаемых.

Рассмотрим, например, S-блок $(x, y + xz, z + xy + xz)$ из класса \mathcal{Q}_2^3 и его разбиение $P(S)$, удовлетворяющее условию неполноты:

$$\begin{aligned} S_1(v) &= (x_2, y_2 + x_2z_2 + x_2z_3 + x_3z_2, z_2 + x_2y_2 + x_2y_3 + x_3y_2 + x_2z_2 + x_2z_3 + x_3z_2); \\ S_2(v) &= (x_3, y_3 + x_3z_3 + x_1z_3 + x_3z_1, z_3 + x_3y_3 + x_1y_3 + x_3y_1 + x_3z_3 + x_1z_3 + x_3z_1); \\ S_3(v) &= (x_1, y_1 + x_1z_1 + x_1z_2 + x_2z_1, z_1 + x_1y_1 + x_1y_2 + x_2y_1 + x_1z_1 + x_1z_2 + x_2z_1). \end{aligned}$$

Условие 2 не выполняется. Однако при добавлении следующей комбинации корректирующих слагаемых (выделены подчеркиванием) разбиение становится допустимым:

$$\begin{aligned} S_1(v) &= (x_2, y_2 + x_2z_2 + x_2z_3 + x_3z_2 + \underline{z_2}, z_2 + x_2y_2 + x_2y_3 + x_3y_2 + x_2z_2 + x_2z_3 + x_3z_2 + \underline{y_3 + z_2}); \\ S_2(v) &= (x_3, y_3 + x_3z_3 + x_1z_3 + x_3z_1 + \underline{z_1}, z_3 + x_3y_3 + x_1y_3 + x_3y_1 + x_3z_3 + x_1z_3 + x_3z_1 + \underline{y_3 + z_1}); \\ S_3(v) &= (x_1, y_1 + x_1z_1 + x_1z_2 + x_2z_1 + \underline{z_1} + \underline{z_2}, z_1 + x_1y_1 + x_1y_2 + x_2y_1 + x_1z_1 + x_1z_2 + x_2z_1 + \underline{z_1} + \underline{z_2}). \end{aligned}$$

Для класса \mathcal{Q}_3^3 допустимое разбиение так и не было найдено, а большой перебор комбинаций корректирующих переменных делает поиск трудным, поэтому в [1] авторы обозначили открытый вопрос: существует ли для S-блоков из класса \mathcal{Q}_3^3 допустимое разбиение?

Пусть $S = (f_1, \dots, f_n)$, $S : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^n$ и $P(S)$ — непосредственное разбиение S-блока S на r частей $S_i = (f_{i1}, \dots, f_{ni})$. Рассмотрим векторную функцию $F = (f_{11}, \dots, f_{n1}, \dots, f_{1r}, \dots, f_{nr})$. Будем говорить, что функция $C_F = (c_{11}, \dots, c_{n1}, \dots, c_{1r}, \dots, c_{nr})$, $c_{ij} : \mathbb{Z}_2^{nr} \rightarrow \mathbb{Z}_2$ — *корректирующая функция* для F , если функция $F + C_F$ обладает следующими свойствами:

- 1) $f_i = \sum_{j=1}^r (f_{ij} + c_{ij})$ для каждого $i = 1, \dots, n$;
- 2) $f_{ij} + c_{ij}$ не зависит от переменных x_{1j}, \dots, x_{nj} для каждого $i = 1, \dots, n$, $j = 1, \dots, r$.

Пусть $k \in \{1, \dots, nr\}$, (i_1j_1, \dots, i_kj_k) — набор индексов длины k из множества $\{11, \dots, n1, \dots, 1r, \dots, nr\}$. Определим множество $C_{i_1j_1, \dots, i_kj_k}^k = \{C_F : (f_{i_1j_1} + c_{i_1j_1}, \dots, f_{i_kj_k} + c_{i_kj_k})$ — сбалансированная функция из \mathbb{Z}_2^{nr} в $\mathbb{Z}_2^k\}$. Пусть $C = \bigcap_k \bigcap_{i_1j_1, \dots, i_kj_k} C_{i_1j_1, \dots, i_kj_k}^k$.

Теорема 2. Функция $F + C_F$ взаимно однозначна, если и только если $C_F \in C$.

Теорема 2 даёт алгоритм отыскания возможного допустимого разбиения, так как для любой функции $C_F \in C$ разбиение $S'_1 = (f_{11} + c_{11}, \dots, f_{n1} + c_{n1}), \dots, S'_r = (f_{1r} + c_{1r}, \dots, f_{nr} + c_{nr})$ по теореме 2 является допустимым. Для S-блока из \mathcal{Q}_3^3 доказано, что множество C пусто. Следовательно, не существует допустимого разбиения данного S-блока, и доказана следующая

Теорема 3. Для S-блоков из класса \mathcal{Q}_3^3 не существует допустимого разбиения.

ЛИТЕРАТУРА

1. *Bilgin B., Nikova S., Nikov V., et al.* Threshold implementations of all 3x3 and 4x4 S-boxes // CHES 2012. LNCS. 2012. V. 7428. P. 76–91.

УДК 056.55

АЛГОРИТМ ВОССТАНОВЛЕНИЯ ОТКРЫТОГО ТЕКСТА ПО ШИФРТЕКСТУ В КРИПТОСИСТЕМЕ МАК-ЭЛИСА

А. К. Калужин, И. В. Чижов

Предлагается алгоритм неструктурной атаки на кодовую криптосистему Мак-Элиса с целью дешифрования сообщения, основывающийся на алгоритме Бернштейна — Ланг — Петерса и работающий быстрее любого другого существующего алгоритма неструктурной атаки. Тем самым сделан ещё один шаг в приближении к нижней оценке сложности таких алгоритмов, доказанной М. Финиазом и Н. Сендрие.

Ключевые слова: *криптосистема Мак-Элиса, неструктурные атаки, алгоритм Бернштейна — Ланг — Петерса, алгоритм Шабо — Канто.*

Рассматриваются неструктурные атаки на криптосистему с открытым ключом Мак-Элиса [1] с целью дешифрования сообщения. По сути, решается уравнение $m \cdot G + e = c$, где m и e неизвестны, а $\text{wt}(e) = t$. При этом m — исходное сообщение, G — порождающая матрица кода (открытый ключ), e — вектор ошибки, c — вектор, который подвергается дешифрованию. Найдя вектор ошибки e , мы решим систему полностью, так как вектор m находится из системы линейных уравнений. Все наилучшие алгоритмы неструктурной атаки на систему Мак-Элиса (Штерна, Шабо — Канто и Бернштейна — Ланг — Петерса) основываются на одной идее: итеративно генерируются различные базисы кода и решается задача в предположении, что вектор ошибки e можно выразить через $2p$ (p — параметр алгоритмов) некоторых из зафиксированных векторов базиса.

В 2009 г. М. Финиаз и Н. Синдреир в работе [2] доказали нижнюю теоретическую оценку ожидаемого количества битовых операций, необходимых для дешифрования сообщения в криптосистеме Мак-Элиса. Для кодов Гошпы (1024, 524, 50) (стандартные параметры криптосистемы Мак-Элиса) эта оценка равна $2^{59,9}$. Оценка идеальна и недостижима (в силу предположений при доказательстве). В то же время ожидаемое количество битовых операций, необходимых для дешифрования сообщения, закодированного с помощью этого кода, составляет:

- 1) для алгоритма Штерна — $2^{66,21}$;
- 2) для алгоритма Шабо — Канто — $2^{64,1}$;
- 3) для алгоритма Бернштейна — Ланг — Петерса — $2^{60,55}$.

То есть существующие алгоритмы уже вплотную приблизились к идеальной оценке ожидаемого количества битовых операций.

В работе представляется модификация алгоритма Бернштейна — Ланг — Петерса [3], которая уменьшает как ожидаемое количество итераций, так и ожидаемое количество битовых операций, выполняемых на одной итерации. Достигается это посредством следующих двух оптимизаций.

- 1) В алгоритме Бернштейна — Ланг — Петерса на каждой итерации фиксируется некоторый базис кода. Он получается из базиса кода, зафиксированного на предыдущей итерации, путём обмена местами s из первых k столбцов матрицы s с s столбцами среди оставшихся, с дальнейшим применением модифицированного преобразования

Гаусса. Оптимизация заключается в том, чтобы не менять столбцы, которые менялись на нескольких предыдущих итерациях; тем самым базисы, фиксируемые на итерациях, становятся более независимыми, что немного повышает вероятность найти вектор ошибки e .

2) На очередной итерации алгоритма Бернштейна — Ланг — Петерса ищется вектор ошибки e в виде линейной комбинации ровно $2p$ векторов базиса. Некоторым образом выбираются линейные комбинации из $2p$ векторов базиса, которые образуют векторы-кандидаты. Если вес какого-то вектора-кандидата равен t , то это и есть искомый вектор e . Для проверки необходимо вычислить веса всех векторов-кандидатов. Бернштейн, Ланг и Петерс предлагают считать вес каждого вектора, пока он не превысит t (далее считать бессмысленно). Оптимизация заключается в том, чтобы отбросить все векторы-кандидаты, у которых среди первых a координат больше чем b координат принимают значение 1. Для остальных осуществить проверку так же, как её делают Бернштейн, Ланг и Петерс. Здесь a и b — новые параметры алгоритма. Смысл оптимизации заключается в следующем: на раннем этапе будет отброшено очень много неподходящих векторов-кандидатов, в то время как подходящий вектор-кандидат может быть отброшен с очень маленькой вероятностью.

Для представленного алгоритма ожидаемое количество битовых операций, необходимых для дешифрования сообщения, закодированного с помощью кодов Гоппы (1024, 524, 50), равно $2^{60,1}$. Это на 27,5% меньше, чем для алгоритма Бернштейна — Ланг — Петерса, самого быстрого из существующих алгоритмов неструктурной атаки на криптосистему Мак-Элиса. Тем самым осуществлено ещё большее приближение к теоретической оценке количества битовых операций при дешифровании сообщения.

ЛИТЕРАТУРА

1. *McEliece R. J.* A public-key cryptosystem based on algebraic coding theory // DSN Progress Report. January and February 1978. No. 42–44. P. 114–116.
2. *Finiasz M. and Sendrier N.* Security bounds for the design of code-based cryptosystems // Asiacrypt'2009. LNCS. 2009. V. 5912. P. 88–105.
3. *Bernstein D. J., Lange T., and Peters C.* Attacking and defending the McEliece cryptosystem // Post-Quantum Cryptography: Second International Workshop, PQCrypto 2008. Cincinnati, OH, USA. October 17–19, 2008. P. 31–46.

УДК 519.17, 004.056.2, 004.056.53

О ВЕРОЯТНОСТНЫХ ХАРАКТЕРИСТИКАХ СЛУЧАЙНЫХ ГРАФОВ, ПОРОЖДАЕМЫХ АЛГОРИТМАМИ ПОИСКА КОЛЛИЗИЙ КРИПТОГРАФИЧЕСКИХ ХЭШ-ФУНКЦИЙ

Г. А. Карпунин

Описывается теоретико-графовая модель некоторых алгоритмов поиска коллизий хэш-функций SHA-1 и RIPEMD, и в данной модели выводится точная формула средней трудоёмкости этих алгоритмов.

Ключевые слова: криптографические хэш-функции, коллизии, случайные графы.

В алгоритмах поиска коллизий некоторых хэш-функций семейства MDx (см., например, SHA-1 [1] и RIPEMD [2, 3]), встречается процедура \mathcal{A} , которую можно смоделировать случайным процессом Γ_N . Данный случайный процесс строит корневое

дерево G максимальной глубины N . При этом некоторые из вершин дерева G оказываются помеченными как плодоносящие, и только плодоносящие вершины могут иметь потомков. Процесс Γ_N считается успешным, если он построит дерево G глубины ровно N и последняя вершина на глубине N , которую сформирует процесс, окажется помеченной как плодоносящая.

У процесса Γ_N имеется два набора параметров $\{p_k\}_{k=N}^0$ и $\{n_k\}_{k=N}^1$, где $0 \leq p_k \leq 1$ и $n_k \in \mathbb{N}$. Формально процесс $\Gamma_N(\{p_k\}_{k=N}^0; \{n_k\}_{k=N}^1)$ описывается с помощью следующей рекурсивной процедуры. На вход процессу подается корневая вершина R . Сначала процесс с вероятностью p_N помечает её как плодоносящую. Если R оказалась непомеченной, то процесс завершает неуспешно свою работу и в качестве построенного графа возвращает лишь непомеченный корень R . Если же R оказалась помеченной как плодоносящая, то процесс строит её первого потомка R_1 и передает его в качестве корневой вершины производному процессу $\Gamma_{N-1}(\{p_k\}_{k=N-1}^0; \{n_k\}_{k=N-1}^1)$. Если производный процесс завершился успешно, то и процесс $\Gamma_N(\{p_k\}_{k=N}^0; \{n_k\}_{k=N}^1)$ также завершается успешно, иначе он строит второго потомка R_2 и для него снова запускает производный процесс $\Gamma_{N-1}(\{p_k\}_{k=N-1}^0; \{n_k\}_{k=N-1}^1)$. Всего может быть построено максимум n_N потомков R_1, \dots, R_{n_N} . Если для каждого из них производный процесс завершился неуспешно, то и процесс $\Gamma_N(\{p_k\}_{k=N}^0; \{n_k\}_{k=N}^1)$ также завершается неуспешно. Пограничный процесс $\Gamma_0(p_0; \emptyset)$ никаких потомков не строит, он просто помечает с вероятностью p_0 корневую вершину как плодоносящую.

Обозначим через P_i вероятность успешного завершения процесса $\Gamma_i(\{p_k\}_{k=i}^0; \{n_k\}_{k=i}^1)$. Несложно показать, что для вероятностей P_i выполняется следующее рекуррентное соотношение:

$$P_i = p_i(1 - (1 - P_{i-1})^{n_i}), \quad (1)$$

при этом из определения очевидно, что $P_0 = p_0$. С помощью формулы (1) можно вычислить вероятность успеха P_N процедуры \mathcal{A} .

На практике [1–3] алгоритмы поиска коллизий запускают процедуру \mathcal{A} несколько раз до первого успеха и в качестве меры эффективности всего алгоритма берётся его средняя трудоёмкость, которая равна величине $\mathbb{E}T(\mathcal{A})/P_N$, где $T(\mathcal{A})$ — трудоёмкость процедуры \mathcal{A} . В качестве меры трудоёмкости процедуры \mathcal{A} , в свою очередь, служит мощность множества вершин $V(G)$ построенного графа G . Таким образом, практически важной величиной является отношение $\mathbb{E}V(G)/P_N$, для вычисления и оценки которого доказана теорема 1.

Теорема 1. Имеют место следующие соотношения:

$$\frac{\mathbb{E}V(G)}{P_N} = \sum_{i=0}^N \frac{1}{P_i} \geq \lim_{\substack{n_k \rightarrow \infty \\ k=1, \dots, N}} \sum_{i=0}^N \frac{1}{P_i} = \sum_{i=0}^N \frac{1}{p_i}.$$

Из теоремы 1 следует, что для минимизации средней трудоёмкости алгоритмов поиска коллизий, использующих процедуру \mathcal{A} , параметры n_k необходимо выбирать как можно больше, если на них нет других ограничений.

ЛИТЕРАТУРА

1. De Cannière C. and Rechberger C. Finding SHA-1 characteristics: general results and applications // ASIACRYPT-2006. LNCS. 2006. V. 4284. P. 1–20.

2. Wang X., Lai X., Feng D., et al. Cryptanalysis of the hash functions MD4 and RIPEMD // EUROCRYPT-2005. LNCS. 2005. V. 3494. P. 1–18.
3. Ермолаева Е. З., Карпунин Г. А. Оценки сложности поиска коллизий для хэш-функции RIPEMD // Прикладная дискретная математика. Приложение. 2012. № 5. С. 43–44.

УДК 519.713

ОБ ОБРАТИМОСТИ КОНЕЧНЫХ АВТОМАТОВ С КОНЕЧНОЙ ЗАДЕРЖКОЙ

Д. А. Катеринский

Построены экспериментальные оценки доли обратимых, слабо обратимых и сильно обратимых конечных автоматов с конечной задержкой, из которых следует, что эта доля мала (до 3%) для автоматов с близкими мощностями их алфавитов состояний и выходных символов и велика (более 80%) для автоматов, у которых выходной алфавит в 4 раза мощнее входного и в 2 раза — внутреннего.

Ключевые слова: конечные автоматы, слабая обратимость, обратимость, анализ обратимости, синтез обратных автоматов, доля обратимых автоматов.

Рассмотрены автоматы, обратимые с нулевой задержкой, и автоматы, слабо или сильно обратимые с конечной задержкой. В первых функция выходов инъективна в каждом состоянии, во вторых входная последовательность восстанавливается с задержкой по выходной последовательности и начальному состоянию, а в третьих — только по выходной последовательности. Для каждого типа обратимости известны тест обратимости и алгоритм построения обратного автомата [1, 2].

В работе сообщается о программной реализации этих тестов и алгоритмов и об экспериментальных оценках доли обратимых автоматов всех типов. Полученные оценки приведены на рис. 1, где на оси абсцисс отмечена доля обратимых автоматов, на оси ординат — значения параметров автоматов, для которых проводилось исследование: m , n и k — мощности соответственно входного, внутреннего и выходного алфавитов автомата. В каждой точке оценки построены усреднением результатов вычислений для 10^4 примеров случайных автоматов. Результаты для доли обратимых автоматов с нулевой задержкой совпадают с теоретическими, вычисленными по формуле

$$d = \frac{(C_k^m \cdot m!)^n}{k^{mn}}.$$

Из рис. 1 видно, что:

- 1) доля обратимых автоматов мала (менее 3%), если мощности входного, внутреннего и выходного алфавитов близки друг к другу или мощности внутреннего и выходного алфавитов меньше мощности входного алфавита;
- 2) доля обратимых автоматов высока (более 80%), если мощность выходного алфавита много больше (более чем в 4 раза) мощности входного алфавита и больше (хотя бы в 2 раза) мощности внутреннего алфавита.

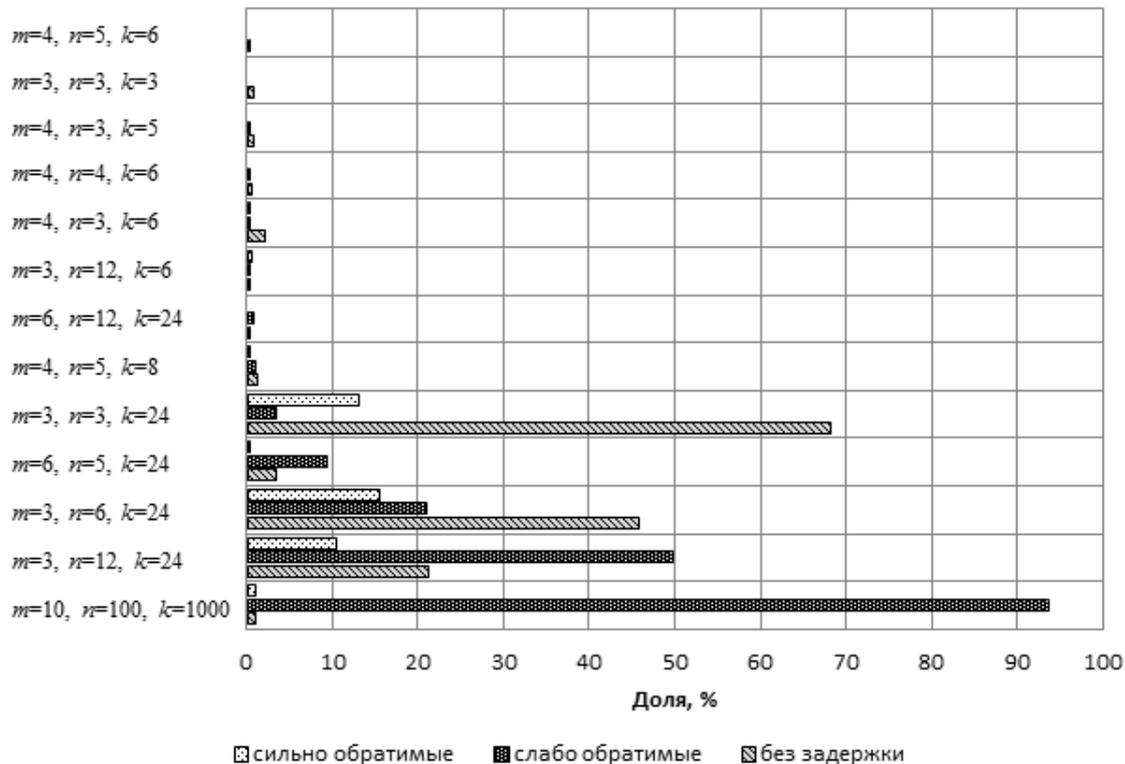


Рис. 1. Доля обратимых автоматов

ЛИТЕРАТУРА

1. *Богаченко Н. Ф.* Применение теоретико-автоматных моделей в криптографии // Математические структуры и моделирование. 2007. Вып. 17. С. 112–120.
2. *Tao R. J.* Finite automata and application to cryptography. Tsinghua: Springer, 2008.

УДК 004.056.55

РЕАЛИЗАЦИЯ НА ПЛИС СИММЕТРИЧНОГО АНАЛОГА FAPKC

Д. С. Ковалев

Рассмотрена реализация на ПЛИС симметричного аналога конечно-автоматной шифрсистемы с открытым ключом (FAPKC). Проведено сравнение ресурсоёмкости и производительности аппаратных реализаций симметричного аналога FAPKC с другими автоматными шифрсистемами. Представлены результаты сравнения ПЛИС-реализаций симметричного аналога FAPKC, AES и других современных блочных шифров.

Ключевые слова: *нелинейный автомат, обратимый с задержкой автомат, конечно-автоматная криптосистема, FAPKC, FASKC, ПЛИС, FPGA, VHDL.*

Данная работа продолжает начатые в [1, 2] исследования конечно-автоматных шифрсистем на пригодность к практическому использованию. Предметом текущего исследования является симметричный аналог конечно-автоматной шифрсистемы с открытым ключом (FAPKC). Критерием оценки пригодности шифра к использованию на практике в данной работе является эффективность его реализации на базе ПЛИС (программируемая логическая интегральная схема).

Идея построения симметричной криптосистемы на основе нелинейных обратимых с задержкой автоматов [3] была предложена в [4]. В симметричном аналоге FAPKC (далее предлагается использовать аббревиатуру FASKC — Finite Automata Single Key Cryptosystem) используются обратимые с задержкой автоматы, т.е. такие автоматы, у которых входное слово восстанавливается по выходному с задержкой на несколько тактов работы, а также автоматы с конечной памятью, значение выходного символа для которых зависит от значений конечного количества входных и выходных символов в предыдущие такты работы. Ключ шифрования состоит из двух обратимых нелинейных автоматов A и B (с небольшой задержкой τ_1 и τ_2 соответственно), обратные к которым могут быть построены с полиномиальной сложностью.

При шифровании к открытому тексту добавляются произвольные $\tau_1 + \tau_2$ символов. После этого находится реакция α автомата A в выбранном начальном состоянии на «расширенный» открытый текст. Шифртекст есть реакция автомата B в выбранном начальном состоянии на входное слово α . Таким образом, длина шифртекста, по сравнению с открытым текстом, увеличивается на $\tau_1 + \tau_2$ символов. При расшифровании сначала находится реакция β автомата, обратного к B , в его начальном состоянии на зашифрованное слово. Исходный открытый текст получается как реакция автомата, обратного к A , в его начальном состоянии на входное слово β , при этом начальные $\tau_1 + \tau_2$ символов отбрасываются. Таким образом, процедуры расшифрования FAPKC и FASKC совпадают.

Криптосистема FASKC описана на языке VHDL и промоделирована в САПР Xilinx WebPack ISE 14.1 на ПЛИС семейств Spartan-3 и Virtex-2. В табл. 1 представлены результаты реализации на Spartan-3 XC3S1000 процедур шифрования и расшифрования криптосистем FAPKC и FASKC. Видно, что процедура шифрования FASKC несколько уступает процедуре шифрования FAPKC как в плане ресурсоёмкости, так и в плане скорости преобразования открытого текста в шифртекст. При этом общая для этих криптосистем процедура расшифрования хотя и использует большее число ресурсов, имеет более высокую производительность.

Т а б л и ц а 1

Сравнение ПЛИС-реализаций FAPKC и FASKC

Процедура	Ресурсоёмкость, Slices (S)	Производительность, Мбит/с (T)	T/S
Шифрование (FAPKC)	3968	294	0,07
Шифрование (FASKC)	4074	252	0,06
Расшифрование	5549	309	0,05

В табл. 2 представлены результаты реализации FASKC на Spartan-3 XA3S1000 и автоматной шифрсистемы Закревского [5] на Spartan-3 XAS400 [6]; производительность FASKC сравнима с производительностью шифра Закревского, в то время как последний имеет в несколько раз меньшую ресурсоёмкость.

Одним из методов изучения эффективности реализации шифрсистем является сравнение исследуемого шифра с криптоалгоритмами, используемыми на практике. В табл. 3 сравниваются реализации FASKC и AES на ПЛИС Spartan-3 и Virtex-2. Результаты реализации AES взяты из работы [7].

Из табл. 3 следует, что хотя FASKC и имеет более высокую производительность, чем AES, ресурсоёмкость последнего меньше в десятки раз. К аналогичным выводам можно прийти, сравнивая FASKC с другими современными блочными шифрами

Т а б л и ц а 2

Сравнение ПЛИС-реализаций FASKC
и шифра Закревского

Шифр	Ресурсоёмкость, Slices (S)	Производительность, Мбит/с (T)	T/S
FASKC (E)	4074	252	0,06
FASKC (D)	5549	309	0,05
Шифр Закревского	1715	290	0,16

Т а б л и ц а 3

Сравнение ПЛИС-реализаций FASKC и AES

Шифр	ПЛИС	Ресурсоёмкость, Slices (S)	Производительность, Мбит/с (T)	T/S
FASKC (E)	XC3S1000-4	4074	252	0,06
FASKC (D)	XC3S1000-4	5549	309	0,05
AES	XC3S50-4	163	208	1,28
FASKC (E)	XC2V1000-6	4082	368	0,09
FASKC (D)	XC2V1000-6	5549	546	0,10
AES	XC2V40-6	146	358	2,45

(3DES, IDEA, CAST), реализованными в [8]. В целом, проведённые исследования показывают, что использование аппаратной реализации симметричного аналога FASKC на практике допустимо, однако возможно не на всех ПЛИС в силу высокой ресурсоёмкости шифрсистемы.

ЛИТЕРАТУРА

1. Ковалев Д. С., Тренькаев В. Н. Реализация на ПЛИС шифра FASKC // Прикладная дискретная математика. Приложение. 2011. № 4. С. 33–34.
2. Ковалев Д. С. Реализация на ПЛИС шифра FASKC-4 // Прикладная дискретная математика. Приложение. 2012. № 5. С. 44–46.
3. Tao R. J. Finite automata and application to cryptography. Tsinghua University Press and Springer, 2008.
4. Abubaker S. Lightweight and secure cryptosystems based on finite automata. Электронные данные. Режим доступа: <http://webdocs.cs.ualberta.ca/~vogt/networks/3-2-Abubaker.pdf>, свободный.
5. Закревский А. Д. Метод автоматической шифрации сообщений // Прикладная дискретная математика. 2009. № 2. С. 127–137.
6. Милошенко А. В. Аппаратная реализация шифрсистемы, основанной на автомате Закревского // Прикладная дискретная математика. Приложение. 2010. № 3. С. 23–24.
7. Rowroy G., Standaert F. X., Quisquater J. J., and Legat J. D. Compact and efficient encryption/decryption module for FPGA implementation of the AES Rijndael very well suited for small embedded applications // Proc. Intern. Conf. Inform. Technology: Coding and Computing. 2004. V. 2. P. 583–587.
8. Kitsos P., Sklavos N., Galanis M. D., and Koufopavlou O. 64-bit Block ciphers: hardware implementations and comparison analysis // Comput. Electric. Eng. 2004. No. 30. P. 593–604.

УДК 519.6

О БЛОЧНЫХ ШИФРАХ, ПОСТРОЕННЫХ НА ОСНОВЕ РЕГИСТРОВ СДВИГА С ДВУМЯ ОБРАТНЫМИ СВЯЗЯМИ

А. М. Коренева

Получены условия биективности и инволютивности алгоритма блочного шифрования, построенного на основе регистра сдвига с двумя обратными связями над пространством двоичных векторов. Построен пример алгоритма блочного шифрования с указанными свойствами на основе регистра сдвига длины 4.

Ключевые слова: биективность, итеративные симметричные блочные шифры, инволютивность алгоритма шифрования, регистры сдвига.

При построении и анализе блочных шифров Фейстеля и более общих моделей регистрового типа необходимо ответить на ряд актуальных вопросов. К таким вопросам относятся определение инволютивности алгоритма шифрования (инволютивность важна для удобства технической и программной реализации), перемешивающих свойств алгоритма (важных с точки зрения противодействия методам последовательного опробования элементов ключа и дифференциального криптоанализа) и др. При усложнении модели алгоритма блочного шифрования необходимо сохранить биективность шифрующих преобразований. В продолжение исследований алгоритмов блочного шифрования, обобщающих шифры Фейстеля [1], в работе рассмотрены алгоритмы, построенные на основе регистров сдвига с двумя обратными связями над пространством двоичных векторов. Такие алгоритмы представляют интерес в связи с тем, что имеют более сильные перемешивающие свойства по сравнению с регистрами с одной обратной связью [2]. Получены условия биективности преобразования регистра сдвига с двумя обратными связями и условия инволютивности соответствующего алгоритма блочного шифрования. На примере алгоритма блочного шифрования на базе регистра сдвига длины 4 показана практическая возможность обеспечения ряда положительных криптографических свойств блочного шифра.

Рассмотрим функции $\varphi_i(x_1, \dots, x_n) : X^n \rightarrow X$, $i = 1, 2$, X — конечное множество. Система функций $\varphi = \{\varphi_1(x_1, \dots, x_n), \varphi_2(x_1, \dots, x_n)\}$ биективна по множеству переменных $\{x_i, x_j\}$, если φ реализует биективное преобразование множества X^2 при любой фиксации переменных $\{x_1, \dots, x_n\} \setminus \{x_i, x_j\}$. Далее X — векторное пространство.

Утверждение 1. Если $\varphi_1 = x_i + f_1(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$, $\varphi_2 = x_j + f_2(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_{j-1}, x_{j+1}, \dots, x_n)$, то система φ биективна по множеству переменных $\{x_i, x_j\}$.

Автономным регистром сдвига длины n над X с обратными связями $\varphi_m(x_1, \dots, x_n)$ и $\varphi_n(x_1, \dots, x_n)$ назовём преобразование g_φ множества X^n , задаваемое системой координатных функций $\{\varphi_1(x_1, \dots, x_n), \dots, \varphi_n(x_1, \dots, x_n)\}$, где $\varphi_i(x_1, \dots, x_n) = x_{i+1}$ для всех $i \in \{1, \dots, n-1\} \setminus \{m\}$, $\varphi = \{\varphi_m(x_1, \dots, x_n), \varphi_n(x_1, \dots, x_n)\}$. Здесь m — параметр регистрового преобразования, $1 \leq m \leq n-1$.

Теорема 1. Преобразование регистра сдвига g_φ биективно, если и только если система φ биективна по множеству переменных $\{x_1, x_{m+1}\}$.

В соответствии с утверждением 1 преобразование g_φ биективно, если при $m < n-1$ имеет место $\varphi_m = x_{m+1} + f_m(x_2, \dots, x_m, x_{m+2}, \dots, x_n)$, $\varphi_n = x_1 + f_n(x_2, \dots, x_n)$ или при $m = n-1$ выполняется $\varphi_m = x_n + f_m(x_2, \dots, x_{n-1})$, $\varphi_n = x_1 + f_n(x_2, \dots, x_n)$.

Пусть $X = V_r$ — множество двоичных r -мерных векторов. Рассмотрим блочный шифр $C(g_{\varphi,q})$ с раундовой подстановкой $g_{\varphi,q}$ множества X^n , задаваемой системой координатных функций $\{\varphi_1(x_1, \dots, x_n, q), \dots, \varphi_n(x_1, \dots, x_n, q)\}$, где $\varphi_i(x_1, \dots, x_n) = x_{i+1}$ для всех $i \in \{1, \dots, n-1\} \setminus \{m\}$, $1 \leq m < n-1$; q — бинарный раундовый ключ; координатные функции φ_m и φ_n имеют вид $\varphi_m = x_{m+1} \oplus f_m(x_2, \dots, x_m, x_{m+2}, \dots, x_n, q)$, $\varphi_n = x_1 \oplus f_n(x_2, \dots, x_n, q)$.

Определим условия инволютивности рассматриваемого шифра — условия, при которых расшифрование отличается от зашифрования только порядком использования раундовых ключей. Обозначим через I_n инволюцию степени n вида $I_n(x_1, \dots, x_n) = (x_n, \dots, x_1)$. Функцию f_n назовем инвариантной относительно инволюции I_{n-1} , если $f_n(x_2, \dots, x_n, q) = f_n(x_n, \dots, x_2, q)$. Функцию f_m назовем инвариантной относительно инволюции I_{n-2} , если $f_m(x_2, \dots, x_m, x_{m+2}, \dots, x_n, q) = f_m(x_n, \dots, x_{m+2}, x_m, \dots, x_2, q)$.

Лемма 1. Если функция f_n инвариантна относительно инволюции I_{n-1} , а функция f_m инвариантна относительно инволюции I_{n-2} , то для раундовой подстановки выполнено равенство $(g_{\varphi,q})^{-1} = I_n g_{\varphi,q} I_n$.

Теорема 2. Пусть h -раундовый блочный шифр $C(g_{\varphi,q})$ при шифровании реализует произведение раундовых подстановок $g_{\varphi,q_1}, \dots, g_{\varphi,q_h}$ и инволюции I_n . Тогда если функция f_n инвариантна относительно инволюции I_{n-1} и функция f_m инвариантна относительно инволюции I_{n-2} , то алгоритм шифрования инволютивен и расшифрование отличается от зашифрования использованием раундовых ключей в обратном порядке.

Пример (инволютивный алгоритм блочного шифрования на основе регистра сдвига с двумя обратными связями). Пусть $n = 4$, $m = 2$, $r = 16$. Рассмотрим алгоритм, который реализует произведение h раундовых подстановок (с раундовыми ключами q_1, \dots, q_h) и инволюции I_4 . Функция усложнения раундовой подстановки представлена парой функций $f_4(x_2, x_3, x_4, q)$ и $f_2(x_2, x_4, q)$. Раундовая подстановка $g_{\varphi,q}$ при использовании ключа q имеет вид $g_{\varphi,q} = (x_2, x_3 \oplus f_2(x_2, x_4, q), x_4, x_1 \oplus f_4(x_2, x_3, x_4, q))$.

Из теоремы 2 следует, что для обеспечения инволютивности рассматриваемого алгоритма шифрования функции $f_4(x_2, x_3, x_4, q)$ и $f_2(x_2, x_4, q)$ должны быть инвариантны относительно инволюций I_3 и I_2 соответственно, иначе говоря, инвариантны относительно перестановки переменных x_2 и x_4 . Рассмотрим варианты построения функций $f_4(x_2, x_3, x_4, q)$ и $f_2(x_2, x_4, q)$.

Используемый 32-битовый раундовый ключ q разобьём на два 16-битовых подключа: $q = (q_1, q_2)$. Тогда указанным требованиям удовлетворяют, в частности, функции вида $f_4(x_2, x_3, x_4, q) = (y_2 \vee y_4) \oplus S_4(y_3)$, $f_2(x_2, x_4, q) = S_2(x_2 \oplus x_4) \oplus q_2$, где $(y_2, y_3, y_4) = (x_2 \oplus q_1, x_3 \oplus q_2, x_4 \oplus q_1)$, \vee — покоординатная дизъюнкция 16-битовых векторов, S_2, S_4 — нелинейные преобразования множества V_{16} (например, четвёрки s -боксов $V_4 \rightarrow V_4$). Применяя h раундов шифрования открытого текста $(x_1(0), x_2(0), x_3(0), x_4(0))$ и инволюцию I_4 , получаем зашифрованный текст в режиме *ECB*, равный $(x_4(h), x_3(h), x_2(h), x_1(h))$. В соответствии с теоремой 2 данный алгоритм шифрования инволютивен, расшифрование отличается от зашифрования использованием раундовых ключей в обратном порядке. Другие положительные криптографические свойства шифра могут быть обеспечены с помощью выбора числа циклов шифрования h , преобразований S_2, S_4 , ключевого расписания и др.

ЛИТЕРАТУРА

1. Коренева А. М., Фомичев В. М. Об одном обобщении блочных шифров Фейстеля // Прикладная дискретная математика. 2012. № 3. С. 34–40.

2. Коренева А. М., Фомичев В. М. Криптографические свойства блочных шифров, построенных на основе регистров сдвига // Прикладная дискретная математика. Приложение. 2012. № 5. С. 49–51.

УДК 519.151, 519.725, 519.165

КОНСТРУКЦИИ ИДЕАЛЬНЫХ СХЕМ РАЗДЕЛЕНИЯ СЕКРЕТА

Н. В. Медведев, С. С. Титов

Работа посвящена исследованию вопросов разграничения доступа к информации при помощи линейных идеальных однородных схем разделения секрета. Приведена конструкция таких схем над любым полем $\text{GF}(q)$. Путём добавления участников показано, что такие схемы сводятся к схемам на проективных пространствах.

Ключевые слова: однородные схемы разделения секрета, структуры доступа, матроиды, код Рида – Маллера, идеальные схемы.

Неотъемлемыми атрибутами современных компьютерных систем и сетей передачи данных являются криптографические протоколы защиты информации. На этом пути часто возникают сложные проблемы, требующие привлечения серьёзного математического аппарата. Одна из таких актуальных и активно исследуемых западными специалистами областей — разграничение доступа [1] при помощи протоколов (схем) разделения секрета (СРС) [2, 3].

Механизм работы СРС заключается в предоставлении участникам долей секрета таким образом, чтобы заранее заданные коалиции участников (разрешённые коалиции) могли однозначно восстановить секрет [4]. Особый интерес вызывают однородные СРС [5–7], которые допускают идеальную реализацию. При этом ограничиваются рассмотрением разделяющих СРС, т. е. таких, где нет незаменимых участников [6].

Разрешённые коалиции идеальной совершенной схемы разделения секрета определяются циклами некоторого связного матроида, изучение которого и даёт структуру доступа [8]. В терминах циклов аксиом всего две. Представляется естественным рассмотреть двойственный вариант аксиоматизации матроида, а именно использовать не циклы C матроида M , а его нуль-множества Z , т. е. $Z = M \setminus C$, которые можно назвать «антициклами». Тогда аксиомы матроида в терминах антициклов имеют следующий вид: 1) нет антицикла в антицикле, т. е. если Z_1, Z_2 — антициклы и $Z_1 \subset Z_2$, то $Z_1 = Z_2$; 2) если $e \in M$, $e \notin Z_1 \cup Z_2$ и Z_1, Z_2 — антициклы, причём $Z_1 \neq Z_2$, то существует такой антицикл Z , что $(\{e\} \cup (Z_1 \cap Z_2)) \subset Z$.

Перейдём к рассмотрению матроидов в проективном m -мерном пространстве M над $\text{GF}(q)$. Возьмём в качестве нуль-множеств Z гиперпространства в M . Как известно [9], $|M| = (q^{m+1} - 1)/(q - 1)$ и $|Z| = (q^m - 1)/(q - 1)$. Поскольку любые два гиперпространства Z_i и Z_j всегда пересекаются, т. е. $(Z_i \cap Z_j) \neq \emptyset$, причём $\dim Z = m - 1$, $\dim(Z_i \cap Z_j) = m - 2$, то для любой точки $e \notin (Z_i \cap Z_j)$ существует единственное гиперпространство Z , натянутое на $\{e\}$ и на пересечение гиперпространств Z_i и Z_j , так что $Z = \langle \{e\}, Z_i \cap Z_j \rangle$. А это — не что иное, как вторая аксиома матроида в терминах антициклов, которую можно назвать усиленной, так как существует единственное такое гиперпространство. Следовательно, вторая аксиома матроида выполняется. Первая аксиома матроида с очевидностью выполняется, так как размерности гиперпространств одинаковы и антицикла в антицикле быть не может.

Далее рассмотрим матроиды в аффинном m -мерном пространстве M над $\text{GF}(q)$. Как известно [9], $|M| = q^m$ и $|Z| = q^{m-1}$. В аффинном пространстве может быть два случая пересечения гиперпространств Z_i и Z_j : 1) либо пересекаются, т. е. $Z_i \cap Z_j \neq \emptyset$, тогда вторая аксиома матроида выполняется, как в проективном пространстве; 2) либо параллельны, т. е. $Z_i \cap Z_j = \emptyset$, тогда это тривиальный случай и вторая аксиома матроида также выполняется, так как объединение двух соответствующих гиперпространств циклов образует всё пространство M . Первая аксиома матроида выполняется в обоих случаях, как и в проективном пространстве.

При реализации идеальной однородной совершенной СРС гиперпространства соответствуют линейным функциям. На основе обобщённых кодов Рида — Маллера получена

Теорема 1. Аффинное и проективное пространства над $\text{GF}(q)$ являются однородными матроидами с гиперпространствами в качестве антициклов.

При этом возникает естественный и сложный вопрос полного описания класса однородных СРС над $\text{GF}(q)$. Для решения этого вопроса рассмотрена возможность добавления участников СРС однородного матроида. Путём комбинаторных рассуждений доказано

Утверждение 1. Линейные однородные разделяющие СРС над $\text{GF}(q)$ сводятся к подсхемам схемы некоторого проективного пространства над $\text{GF}(q)$ с гиперпространствами в качестве антициклов.

ЛИТЕРАТУРА

1. *Гайдамакин Н. А.* Разграничение доступа к информации в компьютерных системах. Екатеринбург: Изд-во Урал. ун-та, 2003.
2. *Shamir A.* How to share a secret // Comm. ACM. NY, USA: ACM, 1979. V. 22. No. 11. P. 612–613.
3. *Черемушкин А. В.* Криптографические протоколы: основные свойства и уязвимости // Прикладная дискретная математика. Приложение. 2009. № 2. С. 115–150.
4. Введение в криптографию / под общ. ред. В. В. Яценко. СПб.: Питер, 2001.
5. *Marti-Farre J. and Padro C.* Secret sharing schemes on sparse homogeneous access structures with rank three // Electronic J. Combinatorics. 2004. No. 11(1). Research Paper 72. 16 p.
6. *Медведев Н. В., Титов С. С.* Бинарные почти пороговые матроиды // Научно-технический вестник Поволжья. 2012. № 4. С. 136–142.
7. *Медведев Н. В., Титов С. С.* Почти пороговые схемы разделения секрета на эллиптических кривых // Доклады ТУСУРа. 2011. № 1(23). Ч. 1. С. 91–96.
8. *Блейкли Г. Р., Кабатянский Г. А.* Обобщенные идеальные схемы, разделяющие секрет, и матроиды // Проблемы передачи информации. 1997. Т. 33. № 3. С. 102–110.
9. *Холл М.* Комбинаторика. М.: Мир, 1970. 424 с.

УДК 519.723

О НЕМИНИМАЛЬНЫХ СОВЕРШЕННЫХ ШИФРАХ

Н. В. Медведева, С. С. Титов

Рассмотрены свойства неминимальных совершенных шифров. Показано, что неминимальный совершенный шифр вкладывается в максимальный совершенный шифр. Доказан аналог теоремы К. Шеннона для неэндоморфных совершенных шифров.

Ключевые слова: совершенные шифры, неэндоморфные шифры, максимальные шифры, неминимальные шифры.

Цель современных криптографических методов защиты информации — создание шифров, не позволяющих раскрыть никаких сведений о соответствующих им открытых текстах. Систематически вопрос о теоретической стойкости шифров впервые исследовал К. Шеннон в своей фундаментальной работе [1], в которой рассмотрена вероятностная модель шифра.

Пусть X, Y — конечные множества открытых и закрытых текстов, с которыми оперирует некоторый шифр замены, K — множество ключей, $|X| = \lambda, |Y| = \mu, |K| = \pi, \lambda > 1, \mu > 1$. Согласно [2, 3], под шифром Σ_B будем понимать совокупность множеств правил зашифрования и расшифрования с заданными распределениями вероятностей на множествах l -грамм открытых текстов, закрытых текстов и ключей.

Шифры, для которых апостериорные вероятности открытых текстов совпадают с их априорными вероятностями, называются *совершенными*. Такие шифры являются абсолютно стойкими к криптоатакам по шифртексту. Совершенным может быть лишь шифр с неограниченным ключом [3]. Шифры, для которых $|X| = |Y|$, называются *эндоморфными* [1]. К. Шеннон полностью описал эндоморфные совершенные шифры с минимальным возможным числом ключей ($|K| = |Y|$).

Согласно [3], шифр Σ_B называется *сильно совершенным*, если он остается совершенным для любого распределения $P(X)$ вероятностей на множестве открытых текстов.

Шифры, для которых $|Y| < |K|$, называются *неминимальными*. Шифры, для которых $|K| = \mu \cdot (\mu - 1) \cdot \dots \cdot (\mu - \lambda + 1)$, то есть шифры, содержащие все инъекции $X \rightarrow Y$, называются *максимальными*. Доказано

Утверждение 1. Неминимальный совершенный шифр вкладывается в максимальный совершенный шифр.

В утверждении 1 имеется в виду не только теоретико-множественное включение, но и теоретико-вероятностное включение в рамках рассматриваемой модели шифра.

Кроме эндоморфных, существуют также *неэндоморфные* ($|X| < |Y|$) шифры с неравновероятными ключами, являющиеся сильно совершенными. Такие шифры обладают дополнительными свойствами, важнейшие из которых — имитостойкость и помехоустойчивость. Изучение неэндоморфных совершенных шифров в общем виде предполагает знание распределения вероятностей $P(X^l)$ на множестве l -грамм алфавита открытых текстов. В работе [4] рассматриваются комбинаторные аспекты проблематики совершенных шифров. В качестве стандартного аппарата исследования распределения вероятностей на l -граммах используются стохастические матрицы и однородные цепи Маркова. При этом проблематика описания совершенных шифров связана с классическими задачами описания статистически неопределённых систем [5]. Справедлива

Теорема 1. Неэндоморфный совершенный шифр является сильно совершенным.

Эквивалентность понятий совершенности и сильной совершенности для неэндоморфных шифров даёт большие возможности для их изучения.

В работе показано, что распределение вероятностей на множествах l -грамм закрытых текстов и ключей, при котором максимальный шифр будет совершенным, можно найти с помощью системы линейных уравнений. Данная система совместна, при этом её неизвестные принадлежат отрезку $[0; 1]$. Поэтому искомое распределение вероятностей в пространстве вероятностей представляет собой некоторое выпуклое тело P^ℓ

(многогранник) в многомерном евклидовом пространстве. Многогранник P^ℓ описан как выпуклая оболочка вершин. Справедливо

Утверждение 2. Если $\ell_1 < \ell_2$, то для соответствующих многогранников P^{ℓ_1} и P^{ℓ_2} выполняется условие $P^{\ell_1} \subset P^{\ell_2}$, то есть многогранники P^ℓ для ℓ -грамм при разных ℓ вложены в друг друга.

Итак, показано, что изучение неэндоморфных шифров сводится к изучению максимальных неэндоморфных шифров. При этом распределения вероятностей на l -граммах могут рассматриваться и как независимые, что характерно для блочных шифров, и как порождённые режимом гаммирования. Справедлива

Теорема 2. Совершенными максимальными шифрами являются шифры с вероятностями ключей $P(K^\ell)$ из многогранника P^ℓ , и только они.

Данная теорема является аналогом теоремы К. Шеннона, доказанным для общего класса неэндоморфных неминимальных шифров.

ЛИТЕРАТУРА

1. Шеннон К. Терия связи в секретных системах // Работы по теории информации и кибернетике. М.: Наука, 1963. С. 333–402.
2. Алферов А. П., Zubov A. Ю., Кузьмин А. С., Черемушкин А. В. Основы криптографии. М.: Гелиос АРВ, 2001.
3. Zubov A. Ю. Совершенные шифры. М.: Гелиос АРВ, 2003.
4. Титов С. С., Гутарин Д. С., Коновалова С. С. и др. Комбинаторные проблемы существования совершенных шифров // Труды ИММ УрО РАН. 2008. Т. 13. № 4. С. 61–73.
5. Медведева Н. В., Тимофеева Г. А. Сравнение линейных и нелинейных методов доверительного оценивания для статистически неопределённых систем // Автоматика и телемеханика. 2007. № 4. С. 51–60.

УДК 519.7

О СВЯЗЯХ МЕЖДУ ОСНОВНЫМИ ПОНЯТИЯМИ РАЗНОСТНОГО АНАЛИЗА ИТЕРАТИВНЫХ БЛОЧНЫХ ШИФРОВ

А. И. Пестунов

Обозначаются некоторые из проблем и несоответствий в существующей терминологии разностного анализа итеративных блочных шифров. Предлагается один из возможных наборов определений, позволяющий сформировать единообразную систему понятий, которая согласована с существующими терминами. Формализованы соответствия между основными понятиями, в частности показано, что в рамках предлагаемого набора определений дифференциал, характеристика и усечённый дифференциал являются усечёнными характеристиками. Формализовано и обобщено понятие объединения дифференциалов и характеристик.

Ключевые слова: терминология, дифференциальный криптоанализ, разностный анализ, блочный шифр, дифференциал, характеристика.

Разностный (дифференциальный) анализ [1] — это распространённый подход к анализу стойкости итеративных блочных шифров, однако несмотря на его популярность и наличие ряда разновидностей [2–5] к настоящему времени не выработана строгая единообразная терминология, его описывающая: основные понятия вводятся либо не строго, либо с использованием авторских определений. Такая ситуация не мешает разраба-

тывать атаки на шифры, но приводит к тому, что одни и те же понятия определяются по-разному даже признанными учёными. Кроме того, отсутствуют формализованные связи между основными понятиями данного метода: характеристики могут называться дифференциалами, дифференциалы могут объединяться с характеристиками, вероятность усечённой характеристики может вычисляться по аналогии с неусечённой, хотя корректность подобных действий строго не обоснована.

Некоторые проблемы теоретического обоснования разностного анализа уже затрагивались в работах отечественных и зарубежных авторов. В [6] предложена модель марковского шифра и сформулирована гипотеза стохастической эквивалентности, в [7] предложена модель для создания шифра, доказуемо стойкого к классическим вариантам разностного и линейного анализа. Работа [8] посвящена изложению разностного анализа в общем виде применительно к произвольным итеративным блочным шифрам с аддитивным раундовым ключом, в работе [9] теоретически исследована вероятность сохранения однобитовой разности после сложения и вычитания. Автор [10] аналитически вычисляет вероятность успеха разностной атаки в зависимости от параметров блочных шифров. Заметим также, что отечественные учёные уже обращались к разработке общей криптографической терминологии, хотя разностный анализ детально не затрагивался [11, 12].

Рассмотрим некоторые вопросы, возникающие при изучении существующей терминологии разностного анализа. Мы не будем классифицировать эти вопросы, а только покажем, что существуют вполне конкретные несоответствия.

Вопрос 1. Входит ли шифр в определения дифференциала и характеристики? Вопрос возникает в связи с тем, что одни авторы определяют дифференциалы и характеристики соответственно как пару или набор блоков (чисел) [1], а другие говорят о разностях открытых и шифрованных текстов [3, 6]. В первом случае дифференциалы и характеристики, состоящие из одинаковых разностей, могут относиться к разным шифрам, а во втором случае подразумевается конкретизация шифра.

Вопрос 2. Входит ли вероятность в определения дифференциалов и характеристик? В исходной работе по разностному анализу [1] характеристика определяется как набор разностей и только затем вводится понятие вероятности характеристики применительно к конкретному шифру. В то же время автор [8] определяет характеристику как последовательность, состоящую и из разностей, и из вероятностей.

Вопрос 3. Является ли объединение дифференциалов характеристикой? Во многих работах авторы строят дифференциалы и характеристики, называя похожие объекты по-разному. Например, в работе [13] объект

$$(a, b, 0, c) \xrightarrow{8 \text{ раундов}} (e, e, 0, 0) \xrightarrow{8 \text{ раундов}} (g, h, f, 0)$$

назван усечённой характеристикой, а объект

$$(0, a, 0, 0) \xrightarrow{8 \text{ раундов}} (0, b, c, d) \xrightarrow{4 \text{ раунда}} (h, h, f, g)$$

— уже усечённым дифференциалом.

Вопрос 4. Можно ли объединять дифференциал и характеристику? В работе [1] даётся определение объединения двух характеристик. В более поздних работах [3, 6] появились понятия дифференциала, а также усечённых дифференциалов и характеристик, однако строгих определений объединения этих объектов не введено. Другими словами, не даются ответы на вопросы о возможности объединения усечённых и неусечённых характеристик и дифференциалов.

Вопрос 5. Как вычислять вероятность объединения усечённых характеристик и дифференциалов? В [6] вводится понятие марковского шифра и показывается, что, согласно уравнению Колмогорова — Чепмена, вероятность характеристики для такого шифра равна произведению вероятностей характеристик, из которых она состоит. Для усечённых характеристик и дифференциалов такого утверждения нет.

Вопрос 6. Является ли дифференциал характеристикой? В работе [6] отмечено, что однораундовые дифференциал и характеристика «совпадают». Отсюда вытекает актуальность установления соответствий между характеристиками и дифференциалами вне зависимости от числа раундов, а также установления соответствий между ними и их усечёнными аналогами.

Список перечисленных вопросов может быть расширен, но и его достаточно, чтобы увидеть существующие проблемы. В настоящей работе предлагается один из возможных наборов определений, позволяющий сформировать единообразную систему понятий, которая согласована с существующими терминами. Формализованы соответствия между основными понятиями, в частности показано, что в рамках предлагаемого набора определений дифференциал, характеристика и усечённый дифференциал являются усечёнными характеристиками. Формализовано и обобщено понятие объединения дифференциалов и характеристик.

Основной идеей, лежащей в основе предлагаемой системы понятий, является использование масок для работы с неизвестными битами в усечённых разностях.

Определение 1. Пусть $I = \{r_0, \dots, r_T\} \subseteq \{0, 1, \dots, R\}$, где $T \leq R$, причём $r_0 = 0$ и $r_T = R$. Пусть также $\Delta = (\delta^{r_0}, \dots, \delta^{r_T})$ и $M = (m^{r_0}, \dots, m^{r_T})$ — упорядоченные множества, где $\delta^r \in \{0, 1\}^s$, $m^r \in \{0, 1\}^s$, $m^r \neq 0$ и $r \in I$. Тогда упорядоченная тройка (I, Δ, M) называется *R-раундовой усечённой характеристикой*. Обозначим её

$$(\delta^{r_0}, m^{r_0}) \xrightarrow{r_1 - r_0 \text{ раундов}} (\delta^{r_1}, m^{r_1}) \xrightarrow{r_2 - r_1 \text{ раундов}} \dots \xrightarrow{r_T - r_{T-1} \text{ раундов}} (\delta^{r_T}, m^{r_T}).$$

В этом определении роли масок играют величины m^r . При $T = 1$ и, следовательно, $r_T = R$ и $I = \{r_0, r_1\}$ усечённая характеристика является усечённым дифференциалом, а при единичных масках усечённые характеристики и дифференциалы являются неусечёнными. Отсюда следует эквивалентность однораундовых характеристик и дифференциалов. Свойство «усечённости» выражается не только в наличии масок, но и в том, что задействованы не все раунды. Часто маски имеют вид, подобный $(0^{32}, 1^{32}, 0^{32}, 1^{32})$, другими словами, все биты определённых подблоков масок равны 0 или 1 одновременно, поэтому усечённые характеристики обычно обозначаются следующим образом:

$$(? , a , ? , b) \rightarrow (? , ? , ? , c) \rightarrow (d , e , ? , f) \rightarrow (? , g , h , ?),$$

где $a, b, c, d, e, f, g, h \in \{0, 1\}^l$, l — размер подблока.

Определение 2. Пусть E — R -раундовый блочный шифр с блоком размера s , (I, Δ, M) — R -раундовая усечённая характеристика и $p \in [0, 1]$. Будем говорить, что шифр E допускает (имеет) усечённую характеристику (I, Δ, M) с вероятностью p , если при случайно и независимо выбранных ключах раундов и блоках открытого текста C^0 и D^0 выполняется равенство

$$\begin{aligned} P((C^{r_1} \oplus D^{r_1}) \& m^{r_1} = \delta^{r_1} \& m^{r_1}, \dots, (C^{r_T} \oplus D^{r_T}) \& m^{r_T} = \\ = \delta^{r_T} \& m^{r_T} | (C^{r_0} \oplus D^{r_0}) \& m^{r_0} = \delta^{r_0} \& m^{r_0}) &= p. \end{aligned}$$

Определение 3. Характеристика $(\widehat{I}, \widehat{\Delta}, \widehat{M})$ присоединима к \widetilde{R} -раундовой характеристике $(\widetilde{I}, \widetilde{\Delta}, \widetilde{M})$, если выполняются равенства $\widetilde{m}^{\widetilde{R}} = \widehat{m}^0$ и $\delta^{\widetilde{R}} \& m^{\widetilde{R}} = \widehat{\delta}^0 \& \widehat{m}^0$.

Определение 4. Пусть $H_1 = (\widetilde{I}, \widetilde{\Delta}, \widetilde{M})$ и $H_2 = (\widehat{I}, \widehat{\Delta}, \widehat{M})$ — соответственно \widetilde{R} - и \widehat{R} -раундовые усечённые характеристики, причём H_2 присоединима к H_1 , тогда объединением H_1 и H_2 назовём характеристику $H = (I, \Delta, M)$, где $I = \{0 = \widetilde{r}_0, \dots, \widetilde{r}_{\widetilde{T}} = \widetilde{R} + \widehat{r}_0, \widetilde{R} + \widehat{r}_1, \dots, \widetilde{R} + \widehat{r}_{\widehat{T}} = \widetilde{R} + \widehat{R}\}$, $\Delta = (\delta^{\widetilde{r}_0}, \dots, \delta^{\widetilde{r}_{\widetilde{T}}}, \delta^{\widehat{r}_1}, \dots, \delta^{\widehat{r}_{\widehat{T}}})$ и $M = (\widetilde{m}^{\widetilde{r}_0}, \dots, \widetilde{m}^{\widetilde{r}_{\widetilde{T}}}, \widehat{m}^{\widehat{r}_1}, \dots, \widehat{m}^{\widehat{r}_{\widehat{T}}})$. По аналогии с операцией конкатенации будем использовать обозначение $H = H_1|H_2$.

Определение 5. Пусть дана последовательность усечённых характеристик H_1, \dots, H_L , для которых выполняется следующее свойство: H_{l+1} может быть присоединена к H_l , $l = 1, \dots, L - 1$. Тогда объединением L характеристик H_1, H_2, \dots, H_L называется характеристика $H = (\dots((H_1|H_2)|H_3)|\dots)|H_L$. Легко видеть, что определённая таким образом операция объединения ассоциативна, поэтому можно использовать запись $H = H_1|H_2|H_3|\dots|H_L$.

Используя введённые определения, легко показать, что любую усечённую характеристику можно представить в виде объединения усечённых дифференциалов. Вычислить вероятность объединения усечённых характеристик можно с использованием уравнения Колмогорова — Чемпена по аналогии с тем, как это сделано в [6] для неусечённых характеристик.

Утверждение 1. Пусть марковский шифр E представим в виде композиции $E = E^1 \circ \dots \circ E^L$, где E_l , $l = 1, \dots, L$, могут быть раундами или композицией раундов. Пусть также E_l допускает некоторую характеристику H_l с вероятностью p_l . Тогда шифр E допускает характеристику $H = H_1|\dots|H_L$ с вероятностью $p_1 \cdot \dots \cdot p_L$.

Из этого утверждения следует очевидная справедливость общепринятого обозначения для характеристик, состоящих из нескольких дифференциалов:

$$\begin{array}{ccccccccccc} \Delta_0 & \xrightarrow{p_1} & \Delta_1 & \xrightarrow{p_2} & \Delta_2 & \xrightarrow{p_3} & \dots & \xrightarrow{p_{L-1}} & \Delta_{L-1} & \xrightarrow{p_L} & \Delta_L; \\ & & \Delta_{\text{in}} & \xrightarrow[p]{\widetilde{R} \text{ раундов}} & \Delta_{\text{mid}} & \xrightarrow[q]{\widehat{R} \text{ раундов}} & & & \Delta_{\text{out}} & & \end{array}$$

ЛИТЕРАТУРА

1. *Biham E. and Shamir A.* Differential cryptanalysis of DES-like cryptosystems // J. Cryptology. 1991. No. 4. P. 3–72.
2. *Пестунов А. И.* Блочные шифры и их криптоанализ // Вычислительные технологии. 2007. Т. 12. Спец. вып. № 4. С. 42–49.
3. *Knudsen L.* Truncated and higher order differentials // LNCS. 1995. V. 1008. P. 196–211.
4. *Biham E., Biryukov A., and Shamir A.* Cryptanalysis of Skipjack reduced to 31 round using impossible differentials // J. Cryptology. 2005. No. 18. P. 291–311.
5. *De Cannière C., Biryukov A, and Preneel B.* An introduction to block cipher cryptanalysis // Proc. IEEE. 2006. V. 94. No. 2. P. 346–356.
6. *Lai X. and Massey J.* Markov ciphers and differential cryptanalysis // LNCS. 1991. V. 547. P. 17–38.
7. *Vaudenay S.* Decorrelation: a theory for block cipher security // J. Cryptology. 2003. No. 16. P. 249–286.

8. Агибалов Г. П. Элементы теории дифференциального криптоанализа итеративных блочных шифров с аддитивным раундовым ключом // Прикладная дискретная математика. 2008. № 1. С. 34–42.
9. Пестунов А. И. О вероятности протяжки однобитовой разности через сложение и вычитание по модулю // Прикладная дискретная математика. 2012. № 4. С. 53–60.
10. Selçuk A. A. On probability of success in linear and differential cryptanalysis // J. Cryptology. 2007. No. 21. P. 131–147.
11. Словарь криптографических терминов / под ред. Б. А. Погорелова и В. Н. Сачкова. М.: МНЦМО, 2006. 94 с.
12. Погорелов Б. А., Черемушкин А. В., Чечета С. И. Об определении основных криптографических понятий // Доклад на конф. «Математика и безопасность информационных технологий», МаБИТ-03, МГУ, 23–24 октября 2003. М., 2003.
13. Knudsen L. R., Robshaw M. J. B., and Wagner D. Truncated differentials and Skipjack // LNCS. 1999. V. 1666. P. 165–180.

УДК 056.55

УЯЗВИМОСТЬ КРИПТОСИСТЕМЫ МАК-ЭЛИСА, ПОСТРОЕННОЙ НА ОСНОВЕ ДВОИЧНЫХ КОДОВ РИДА — МАЛЛЕРА

И. В. Чижов, М. А. Бородин

Предлагается новый алгоритм восстановления секретного ключа по открытому для криптосистемы Мак-Элиса, построенной на основе двоичных кодов Рида — Маллера $RM(r, m)$. В случае, если значение $d = (r, m - 1)$ ограничено, алгоритм имеет полиномиальную сложность $O(n^d + n^4 \log_2 n)$, где $n = 2^m$. Практические результаты показывают, что предложенная атака позволяет осуществить взлом криптосистемы Мак-Элиса, построенной на основе двоичного кода Рида — Маллера длины $n = 65526$ битов, менее чем за 7 ч на персональном компьютере.

Ключевые слова: *криптосистема Мак-Элиса, коды Рида — Маллера, полиномиальная сложность атаки.*

Рассматривается криптосистема Мак-Элиса, оригинальная версия которой использует коды Гошпы [1]. В 1994 г. В. М. Сидельников в работе [2] предложил использовать для построения криптосистемы Мак-Элиса коды Рида — Маллера, которые позволяют увеличить скорость расшифрования и передачи криптограммы.

На сегодняшний день самым успешным из опубликованных алгоритмов восстановления секретного ключа по открытому для криптосистемы Мак-Элиса, основанной на двоичных кодах Рида — Маллера $RM(r, m)$, является алгоритм Л. Миндера и А. Шокроллахи, предложенный ими в 2007 г. в [3]. Этот алгоритм имеет субэкспоненциальную сложность, его идея заключается в сведении задачи взлома криптосистемы с параметрами кода $RM(r, m)$ к такой же задаче, но для кода $RM(1, m)$. В данной работе предложен другой алгоритм сведения, который имеет полиномиальную сложность для некоторого подмножества кодов Рида — Маллера.

Полученные теоретические результаты можно кратко представить в виде теорем.

Теорема 1. Пусть $(r, m - 1) = 1$. Тогда существует алгоритм со сложностью $O(n^4 \log_2 n)$ битовых операций, который по порождающей матрице кода $RM^\sigma(r, m)$ находит перестановку σ' , такую, что $RM^{\sigma \cdot \sigma'}(r, m) = RM(r, m)$.

И обобщение теоремы 1:

Теорема 2. Пусть $(r, m - 1) = d > 1$. Тогда существует алгоритм со сложностью $O(n^d + n^4 \log_2 n)$ битовых операций, который по порождающей матрице кода $RM^\sigma(r, m)$ находит перестановку σ' , такую, что $RM^{\sigma \cdot \sigma'}(r, m) = RM(r, m)$.

Это означает, что для параметров кода $RM(r, m)$, у которых $(r, m - 1)$ ограничен, предложенная атака имеет полиномиальную сложность.

Теоретические результаты подтверждаются практическими исследованиями (табл. 1 и 2): алгоритм реализован программно и запускался на ноутбуке с процессором 2,1 ГГц. Для тех параметров криптосистемы, когда применение алгоритма не даёт асимптотического ускорения, используется символ «М». Если алгоритм сводит исходную задачу (r, m) к задаче с меньшей трудоёмкостью (d, m) , то это отмечено в табл. 2 символами (d, m) .

Т а б л и ц а 1
**Результат Л. Миндера и
 А. Шокроллахи (процессор 2,4 ГГц)**

r	m			
	8	9	10	11
2	0,04 с	0,24 с	12,14 с	1,77 с
3	0,18 с	1,26 с	16,5 с	5 м 20 с
4		2 м 57 с	22 ч 50 м	10 д 11 ч 55 м

Т а б л и ц а 2
Наш результат (2,1 ГГц)

r	m								
	8	9	10	11	12	13	14	15	16
2	0,007 с	М	0,48 с	М	6 с	М	3 м 13 с	М	2 ч 30 м
3	0,01 с	0,2 с	М	1,35 с	19 с	М	5 м 29 с	30 м 31 с	М
4	0,043 с	М	0,43 с	(2,11)	15 с	М	7 м 10 с	(2,15)	3 ч 28 м
5	0,042 с	0,4 с	0,8 с	М	16,5 с	2 м 1 с	14 м 12 с	53 м	М
6		(2,9)	(3,10)	(2,11)	23 с	М	9 м 28 с	14 м 16 с	(3,16)
7			0,86 с	3,2 с	25 с	3 м 16 с	10 м 54 с	М	6 ч 43 м

ЛИТЕРАТУРА

1. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. Теория кодов, исправляющих ошибки. М.: Связь, 1979.
2. Сидельников В. М. Открытое шифрование на основе двоичных кодов Рида — Маллера // Дискретная математика. 1994. Т. 6. № 2. С. 3–20.
3. Minder L. and Shokrollahi A. Cryptanalysis of the Sidelnikov cryptosystem // LNCS. 2007. V. 4515. P. 347–360.

Секция 3

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ
БЕЗОПАСНОСТИ И НАДЕЖНОСТИ ВЫЧИСЛИТЕЛЬНЫХ
И УПРАВЛЯЮЩИХ СИСТЕМ

УДК 519.718

ОБ ОЦЕНКАХ НЕНАДЕЖНОСТИ СХЕМ
ПРИ ИНВЕРСНЫХ НЕИСПРАВНОСТЯХ
И ОТКАЗАХ ФУНКЦИОНАЛЬНЫХ ЭЛЕМЕНТОВ¹

М. А. Алехина, О. Ю. Барсукова

Рассматривается реализация булевых функций схемами из ненадёжных функциональных элементов в произвольном полном базисе B . Предполагается, что все элементы схемы независимо друг от друга переходят в неисправные состояния двух типов (инверсные неисправности и отказ элементов). Получены верхние и нижние асимптотические оценки ненадёжности схем.

Ключевые слова: булевы функции, функциональный элемент, схема, ненадёжность схемы, инверсные неисправности на выходах элементов, отказ элемента.

Рассмотрим реализацию булевых функций неприводимыми [1] схемами из ненадёжных функциональных элементов в произвольном полном конечном базисе B . Будем считать, что неприводимая схема из ненадёжных элементов реализует функцию $f(x)$ ($x = (x_1, \dots, x_n)$), если при поступлении на входы схемы набора a при отсутствии неисправностей в схеме на её выходе появляется значение $f(a)$. Предполагается, что каждый элемент схемы независимо от других элементов переходит в неисправное состояние первого или второго типа. Неисправность первого типа (отказ элемента) появляется на любом входном наборе элемента с вероятностью $\delta \in (0, 1/2)$ и характеризуется тем, что элемент не работает. Считаем, что в этом случае вся схема не работает и значение на её выходе не определено. Неисправность второго типа характеризуется тем, что в исправном состоянии базисный элемент реализует приписанную ему функцию φ , а в неисправном состоянии — функцию $\bar{\varphi}$, т.е. неисправности второго типа — инверсные неисправности на выходах элементов. Эти неисправности также статистически независимы, появляются независимо друг от друга с вероятностью $\varepsilon \in (0, 1/2)$.

Таким образом, если E_φ — базисный элемент с функцией $\varphi(x_1, \dots, x_m)$, подверженный рассматриваемым неисправностям, то на любом входном наборе (a_1, \dots, a_m) вероятность отказа элемента равна δ ; вероятность появления значения $\bar{\varphi}(a_1, \dots, a_m)$ (ошибки) на выходе элемента равна $\varepsilon(1 - \delta)$; вероятность появления правильного значения $\varphi(a_1, \dots, a_m)$ равна $(1 - \varepsilon)(1 - \delta)$.

Пусть $f(x)$ — произвольная булева функция, а S — любая схема, реализующая $f(x)$. Пусть a — произвольный входной набор схемы S , $b = f(a)$ ($b \in \{0, 1\}$). Обозначим через $P_{\text{отк}}(S, a)$ и $P_{\bar{b}}(S, a)$ соответственно вероятности отказа схемы S и появления \bar{b} (ошибки) на выходе схемы S при входном наборе a . Очевидно, что вероятность $P_{\text{отк}}(S, a)$ не

¹Исследование поддержано грантами РФФИ, проекты № 11-01-00212, 12-01-31340.

зависит ни от входного набора a , ни от структуры схемы S , а зависит только от числа $|S|$ элементов в схеме S : $P_{\text{отк}}(S, a) = 1 - (1 - \delta)^{|S|}$. Поэтому далее вместо $P_{\text{отк}}(S, a)$ будем писать $P_{\text{отк}}(S)$, называя вероятность $P_{\text{отк}}(S)$ *вероятностью отказа схемы S* . *Ненадёжностью схемы S* будем называть число $P_{\varepsilon, \delta}(S) = \max_a \{P_{\bar{b}}(S, a)\}$, где максимум берётся по всем входным наборам a схемы S .

Цель работы — получить верхние и нижние асимптотические оценки ненадёжности. Чтобы сформулировать полученные результаты (теорема 1), введём необходимые и ранее известные определения для случая $\delta = 0$.

Пусть $P_{\varepsilon, 0}(f) = \inf P_{\varepsilon, 0}(S)$, где инфимум берётся по всем схемам S из ненадёжных элементов, реализующим булеву функцию f . Схема A из ненадёжных элементов, реализующая функцию f , называется *асимптотически оптимальной по надёжности*, если $P(A) \sim P_{\varepsilon, 0}(f)$ при $\varepsilon \rightarrow 0$, т. е. $\lim_{\varepsilon \rightarrow 0} \frac{P_{\varepsilon, 0}(f)}{P(A)} = 1$.

Пусть f — произвольная булева функция. Число $k_B(f)$ будем называть *коэффициентом ненадёжности функции f* , если выполняются два условия одновременно: 1) функцию f в базисе B можно реализовать схемой с ненадёжностью, асимптотически не больше $\varepsilon k_B(f)$ при $\varepsilon \rightarrow 0$, и 2) ненадёжность любой схемы, реализующей f в базисе B , асимптотически не меньше чем $\varepsilon k_B(f)$ при $\varepsilon \rightarrow 0$.

Например, $k_B(x_i) = 0$, ($i \in \{1, \dots, n\}$); $k_B(\varphi) = 1$, если φ — одна из функций базиса B . Очевидно, что коэффициент ненадёжности функции зависит только от базиса и самой функции. Известно [2], что в любом полном конечном базисе B для любой функции f число $k_B(f) \in \{0, 1, 2, 3, 4, 5\}$.

Коэффициентом ненадёжности базиса B будем называть число $k_B = \max k_B(f)$, где максимум берётся по всем булевым функциям f .

Очевидно, что при инверсных неисправностях на выходах элементов в любом базисе B любая схема, содержащая хотя бы один элемент, имеет ненадёжность не меньше ε , т. е. $k_B \geq 1$. Поэтому $k_B \in \{1, 2, 3, 4, 5\}$. Например, известно, что если полный базис B содержит функцию голосования, то $k_B = 1$. Кроме того [2], если 1) $B = \{xy, x \oplus y, 1\}$, то $k_B = 2$; 2) $B = \{\bar{x}y, \bar{x} \vee y\}$, то $k_B = 3$; 3) $B = \{\bar{x}y, \bar{x}\}$, то $k_B = 4$; 4) $B = \{xy, \bar{x}\}$, то $k_B = 5$.

Пусть B — произвольный полный конечный базис, а K — множество функций f , для которых $k_B(f) = k_B$. Справедлива

Теорема 1. В базисе B любую функцию f можно реализовать такой неприводимой схемой S , что $P_{\varepsilon, \delta}(S) \lesssim \varepsilon k_B (1 - \delta)^{|S|}$ при $\varepsilon \rightarrow 0$; и для любой функции $h \in K$ и любой схемы A , реализующей h , верно неравенство $P_{\varepsilon, \delta}(A) \gtrsim \varepsilon k_B (1 - \delta)^{|A|}$ при $\varepsilon \rightarrow 0$.

ЛИТЕРАТУРА

1. Лупанов О. Б. Асимптотические оценки сложности управляющих систем. М.: Изд-во Моск. ун-та, 1984.
2. Васин А. В. Асимптотически оптимальные по надёжности схемы в полных базисах из трехвыходовых элементов: дис. ... канд. физ.-мат. наук. Пенза, 2010.

УДК 004.75, 004.78, 608.4

СЕРВИС BLACKBOX ДЛЯ ПРОВЕДЕНИЯ СОРЕВНОВАНИЙ ПО ЗАЩИТЕ КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ CAPTURE THE FLAG

Н. И. Анисеня, Д. А. Стефанцов, Т. А. Торгаева

Представляется разработанная командой Томского госуниверситета SiBears система BlackBox для проведения соревнований по защите компьютерной информации Capture the Flag, основанных на выполнении заданий. Описывается функциональность системы, особенности её разработки и администрирования.

Ключевые слова: *SiBears, BlackBox, CTF.*

Широко распространённые в настоящий момент соревнования по защите компьютерной информации Capture the Flag (CTF) [1] являются хорошим дополнением к программе обучения студентов по специальности «Компьютерная безопасность», развивают навыки поиска уязвимостей и защиты от них. Существует несколько форм проведения таких соревнований, основными из которых являются две: 1) командам участников предлагается набор заданий, классифицированных по сложности и по области знаний, используемых при решении; 2) локальные сети команд участников объединяются в игровую сеть с помощью технологии VPN, после чего участники пытаются скомпрометировать серверы других команд и защитить свои серверы. Первая форма используется, в основном, для проведения отборочных соревнований, а вторая — для проведения финалов соревнований.

Команда Томского госуниверситета SiBears [2] принимает активное участие в играх CTF и является организатором нескольких подобных соревнований. Описание программной системы, разработанной SiBears для проведения соревнований второго типа, можно прочесть в [3]. В данной работе описывается программная система BlackBox [4], разработанная командой SiBears, для проведения соревнований первого типа. Разработка BlackBox начата в 2010 г. выпускником кафедры защиты информации и криптографии Томского госуниверситета Алексеем Краснощёвым и студентом той же кафедры Максимом Цоем. В настоящее время поддержкой и модификацией системы занимаются авторы работы. На базе BlackBox проведены соревнования SiBCTF-2011, отборочный тур соревнований RuCTF 2012 [5], соревнования по защите компьютерной информации для школьников School CTF в 2010, 2011 и 2012 гг.

Соревнование в системе BlackBox представляет собой набор заданий, каждое из которых имеет название, текстовое описание и одно или несколько правильных решений, которые могут быть представлены в виде текста. Дополнительно к условию задания могут прилагаться файлы и может предоставляться доступ к виртуальной машине (VM), образ которой подготовлен автором задания. Полученный в результате решения задания ответ участник вводит в специальное поле на странице задания, после чего система проверяет корректность этого решения. Если решение признано системой корректным, соответствующему участнику начисляются игровые баллы. Если участник входит в команду, которая участвует в соревновании, частью которого является решённое задание, команде также начисляются баллы.

В настоящий момент в BlackBox существуют две системы подсчёта баллов, полученных за решение заданий во время соревнования: с фиксированным количеством баллов за задание и с переменным. В первом случае автор задания самостоятельно определяет количество баллов, которые начисляются участнику или команде за

правильное решение. Во втором случае каждое задание в соревновании оценивается в 100 баллов, которые распределяются поровну между всеми участниками или командами, решившими его. Таким образом, простые задания, которые решены большим количеством участников, принесут каждому из них мало баллов, и наоборот.

По задумке авторов, сервисом BlackBox может воспользоваться любой желающий для проведения своих собственных соревнований. Поэтому в дополнение к стандартным ролям пользователей системы («наблюдатель», «участник соревнования» и «администратор») реализованы роли «автор соревнования» и «автор задания». Для удобства участников была также введена роль «капитан команды» — специализация роли «участник соревнования». На рис. 1 изображена диаграмма UML вариантов использования системы BlackBox, где показано соответствие между ролями пользователей и функциями системы. Варианты использования, доступные некоторой роли, показаны овалами и соединяются с этой ролью сплошной линией. Стрелки между видами пользователей проводятся от более специализированного вида к более общему. Это означает, что, например, участнику соревнования доступны все функции, доступные наблюдателю, и некоторые дополнительные. Варианты использования на диаграмме обозначены номерами из следующего списка:

- 1) регистрация в системе и редактирование информации о себе;
- 2) просмотр рейтингов пользователей и команд — включает в себя просмотр рейтинга команды или пользователя в пределах соревнования или задания или сквозного рейтинга по всем соревнованиям и заданиям;
- 3) просмотр достижений пользователей и команд — достижения отмечаются при определённых условиях, таких, как победа в соревновании, решение первым трёх задач подряд в соревновании, наименьшее количество неправильных ответов в соревновании;
- 4) просмотр журнала соревнования и подсчёт рейтинга по нему — пользователям предоставляется возможность посмотреть журнал событий, произошедших в соревновании;
- 5) решение заданий из архива — после проведения соревнования задачи из него попадают в архив, который может использоваться пользователями для тренировки;
- 6) комментирование страниц — на веб-страницах сервиса можно оставлять комментарии с уточняющими вопросами к жюри;
- 7) решение заданий во время соревнования — за решение заданий во время соревнования участникам или командам начисляются баллы;
- 8) управление командой — капитан команды может пригласить участников в неё, участники могут согласиться или отказаться;
- 9) создание соревнований и управление ими — включает в себя определение названия, описания, способа вычисления рейтинга, времени проведения и группы авторов заданий соревнования;
- 10) создание и редактирование заданий — включает в себя определение названия, описания, дополнительных файлов, образа VM и корректного ответа задания; вместо корректного ответа автор задания может предоставить программу для проверки ответов пользователей на корректность;
- 11) просмотр, редактирование и удаление пользователей, соревнований и заданий — администратору системы предоставляется возможность корректировки любых данных системы.

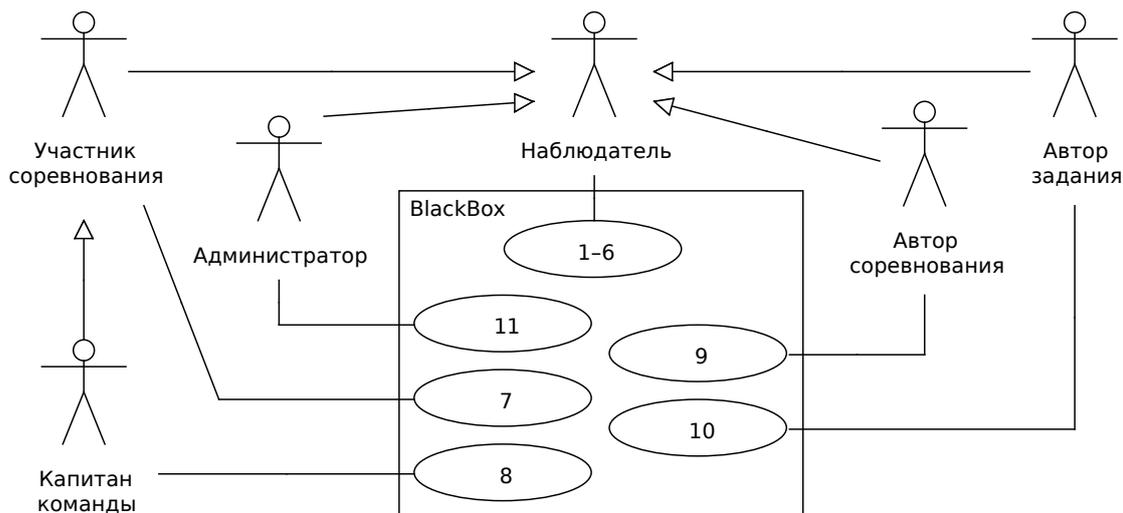


Рис. 1. Диаграмма вариантов использования сервиса BlackBox

Большая часть системы BlackBox реализована на языке программирования Python [6] с использованием библиотеки Django [7]. Основные сложности в реализации системы связаны, во-первых, с предоставлением всем пользователям возможности создания заданий и, во-вторых, с необходимостью синхронизации множества событий, происходящих в системе одновременно.

Поскольку авторами заданий в системе BlackBox могут стать пользователи, которые являются участниками других соревнований в этой же системе, необходимо реализовать механизмы защиты от возможных атак на систему с целью узнать ответы на задания или вызвать отказ системы в обслуживании. При составлении задания автору предоставляется возможность загрузки программы для проверки корректности ответов, присылаемых пользователями. Такая возможность необходима в заданиях, предполагающих множество правильных ответов. Загруженная автором задания программа запускается на одном из серверов системы BlackBox и при этом не является доверенной. Для предотвращения нежелательных действий программы во время выполнения применяется модуль AppArmor [8] подсистемы LSM ядра Linux. Для ограничения программы по времени работы и объёму доступной памяти применяется механизм ограничения ресурсов, выделяемых ядром Linux [9].

Задания могут предполагать доступ участников к некоторому серверу в процессе решения. Например, задание на реализацию атаки типа SQL-инъекция может требовать для решения доступа к веб-серверу, подготовленному автором задания. Система BlackBox предоставляет возможность подготовки образов VM, которые будут запущены во время решения соответствующих заданий. Реализация этой функциональности осуществляется с помощью технологии OpenStack [10], позволяющей организовать управление созданием и запуском VM на нескольких серверах. OpenStack имеет программный интерфейс на языке Python, что позволяет легко интегрировать управление VM в пользовательский веб-интерфейс. Для получения доступа к VM для решения некоторого задания пользователь подключается к сети VM с помощью протокола OpenVPN.

Проблема синхронного доступа к базе данных решена частично с помощью схемы, допускающей модификацию данных только инструкциями типа INSERT [11], частично — использованием библиотеки Celery [12]. Для оптимизации выполнения часто

приходящих одинаковых запросов, например на просмотр рейтинга, применяется кеширование результатов с помощью memcached [13].

Система BlackBox состоит из нескольких крупных компонент: пользовательского веб-интерфейса, обработчиков пользовательских запросов, средства управления базами данных и нескольких баз данных, сервера для управления VM и нескольких серверов VM. Каждая из этих компонент может быть установлена на отдельный компьютер для повышения производительности системы при большом количестве участников, соревнований и заданий. Некоторые компоненты допускают установку на несколько компьютеров. На рис. 2 изображена диаграмма UML развёртывания системы BlackBox: объёмными прямоугольниками изображены возможные отдельные физические компьютеры, на которые устанавливаются компоненты; сами компоненты изображены прямоугольником со стереотипом «component». Система BlackBox допускает масштабирование: элементы диаграммы, помеченные стереотипом «повторяемый», могут присутствовать в системе в нескольких экземплярах. Для ускорения обработки большого числа пользовательских запросов можно увеличить количество компьютеров с обработчиками запросов, а для увеличения количества одновременно работающих VM — увеличить количество серверов VM. При этом один сервер VM может одновременно исполнять более одной VM.

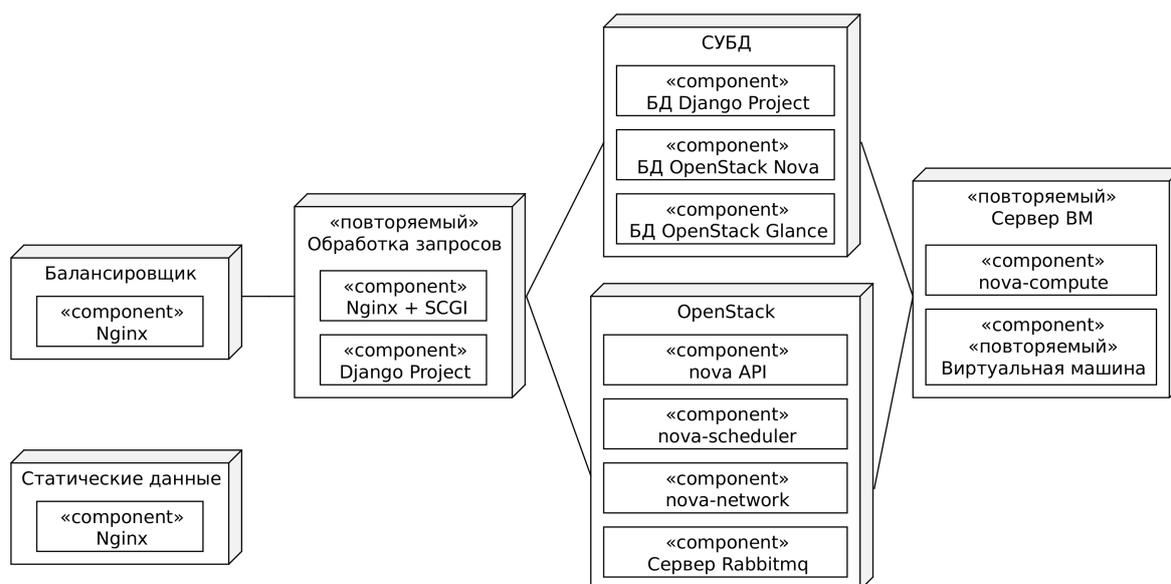


Рис. 2. Диаграмма развёртывания BlackBox

В настоящее время сервис BlackBox удовлетворяет большинству требований, предъявляемых к системам для проведения соревнований CTF, основанных на заданиях. Развитие сервиса видится в его популяризации проведением новых соревнований с большим числом участников, а также интеграцией в него системы для проведения соревнований CTF, основанных на компрометации и защите участниками собственных серверов.

ЛИТЕРАТУРА

1. Колегов Д. Н., Чернушенко Ю. Н. О соревнованиях CTF по компьютерной безопасности // Прикладная дискретная математика. 2008. № 2(2). С. 81–83.
2. <http://sibears.ru/> — Команда Томского государственного университета SiBears. 2013.

3. Ткаченко Н. О., Чернов Д. В. Разработка и реализация сервера игры CTF // Прикладная дискретная математика. Приложение. 2010. № 3. С. 62–64.
4. <http://blackbox.sibears.ru/> — Система для проведения соревнований по защите компьютерной информации. 2013.
5. <http://ructf.org/2012> — Всероссийские межвузовские соревнования по защите информации RuCTF 2012. 2012.
6. <http://python.org/> — Python Programming Language — Official Website. 2013.
7. <https://www.djangoproject.com/> — Django. The Web framework for perfectionists with deadlines. 2013.
8. <http://wiki.apparmor.net> — AppArmor security wiki project. 2013.
9. <http://www.linux.com/learn/docs/man/4047-setrlimit2> — Linux Programmer's Manual (2). getrlimit, setrlimit — get/set resource limits. 2008.
10. <http://www.openstack.org/> — OpenStack. Open source cloud computing software. 2013.
11. <https://www.simple-talk.com/sql/database-administration/database-design-a-point-in-time-architecture/> — Database Design: A Point in Time Architecture. 2007.
12. <http://www.celeryproject.org/> — Celery. Distributed Task Queue. 2012.
13. <http://memcached.org/> — Memcached. A distributed memory object caching system. 2012.

УДК 519.718

О БАЗИСАХ С КОЭФФИЦИЕНТОМ НЕНАДЁЖНОСТИ 1¹

А. В. Васин

Рассматривается реализация булевых функций схемами из ненадёжных функциональных элементов в произвольном полном базисе B . Предполагается, что все элементы схемы независимо друг от друга с вероятностью $\varepsilon \in (0, 1/2)$ подвержены инверсным неисправностям на выходах. Найдено множество G функций, таких, что для почти всех функций ненадёжность асимптотически оптимальных по надёжности схем в базисе B , содержащем функции множества G , равна ε (при $\varepsilon \rightarrow 0$).

Ключевые слова: *ненадёжные функциональные элементы, асимптотически оптимальные по надёжности схемы, инверсные неисправности на выходах элементов.*

Рассматривается реализация булевых функций схемами [1] из ненадёжных функциональных элементов в произвольном полном конечном базисе B . Предполагаем, что все элементы схемы независимо друг от друга с вероятностью $\varepsilon \in (0, 1/2)$ подвержены инверсным неисправностям на выходах. Эти неисправности характеризуются тем, что в исправном состоянии функциональный элемент реализует приписанную ему булеву функцию ψ , а в неисправном — функцию $\bar{\psi}$. Считаем, что схема S из ненадёжных элементов реализует булеву функцию $f(x_1, x_2, \dots, x_n)$, если при поступлении на входы схемы двоичного набора $a = (a_1, a_2, \dots, a_n)$ при отсутствии неисправностей на выходе схемы S появляется значение $f(a)$.

Ненадёжностью $P(S)$ схемы S назовем максимальную вероятность ошибки на выходе схемы S при всевозможных входных наборах схемы. Надёжность схемы S равна $1 - P(S)$. Пусть $P_\varepsilon(f) = \inf P(S)$, где инфимум берется по всем схемам S из ненадёж-

¹Работа выполнена при финансовой поддержке РФФИ, проект № 12-01-31340.

ных элементов, реализующим булеву функцию $f(x_1, x_2, \dots, x_n)$. Схема A из ненадёжных элементов, реализующая функцию f , называется асимптотически оптимальной (асимптотически наилучшей) по надёжности, если $P(A) \sim P_\varepsilon(f)$ при $\varepsilon \rightarrow 0$.

Число k будем называть коэффициентом ненадёжности базиса, если все функции в этом базисе можно реализовать схемами с ненадёжностью, асимптотически не больше $k\varepsilon$ (при $\varepsilon \rightarrow 0$), и найдётся функция f , которую нельзя реализовать схемой с ненадёжностью, асимптотически меньше чем $k\varepsilon$ (при $\varepsilon \rightarrow 0$).

Задача построения асимптотически оптимальных по надёжности схем при инверсных неисправностях на выходах элементов в полных базисах из трёхходовых элементов решена в [2]. В этой работе найден широкий класс базисов, для которых $k = 1$.

Обозначим через K множество функций, для которых асимптотически оптимальные по надёжности схемы в базисе B функционируют с ненадёжностью, асимптотически равной $k\varepsilon$ (при $\varepsilon \rightarrow 0$), где k — коэффициент ненадёжности базиса B .

Нетрудно проверить, что при инверсных неисправностях на выходах элементов в любом базисе ненадёжность любой схемы, содержащей хотя бы один элемент, не меньше ε . Поэтому коэффициент ненадёжности любого базиса не меньше 1.

Для реализации любой функции, отличной от x_i ($i = 1, 2, \dots, n$), требуется не менее одного функционального элемента. Обозначим через $K(n)$ множество булевых функций $K(n) = P_2 \setminus \{x_i : i = 1, \dots, n\}$. Очевидно, что $\frac{|K(n)|}{2^{2^n}} \rightarrow 1$ при $n \rightarrow \infty$. Тогда в случае $k = 1$ множество K равно $K = \bigcap_{i=1}^{\infty} K(n)$.

Замечание 1. Для любой булевой функции $f \in K$ верно $P_\varepsilon(f) \geq \varepsilon$.

Опишем множество G функций $\varphi(x_1, \dots, x_r)$, обладающих свойством повышения надёжности.

Пусть $e_1, e_2, \dots, e_r \in \{0, 1\}^r$, где e_i — вектор, имеющий ровно одну ненулевую компоненту на i -м месте, $i = 1, 2, \dots, r$. Зададим множество E^k , состоящее из r векторов $\hat{e}_1, \hat{e}_2, \dots, \hat{e}_r \in \{0, 1\}^r$, следующим образом: 1) $\hat{e}_i = e_i, i = 1, 2, \dots, k$; 2) $\hat{e}_i = e_i + \sum_{j=1}^{i-1} \lambda_{ij} e_j$, где $\lambda_{ij} \in \{0, 1\}, i = k + 1, k + 2, \dots, r$. Пусть $\varphi(x_1, x_2, \dots, x_r) \in G_0$, если существуют такие двоичные наборы $\tilde{\alpha}, \tilde{\beta}$, что 1) $\varphi(\tilde{\alpha}) = 0, \varphi(\tilde{\beta}) = 1$; 2) для любого набора $\tilde{x} = \tilde{\alpha} + \hat{e}_i$ верно $\varphi(\tilde{x}) = 0$; 3) для любого набора $\tilde{x} = \tilde{\beta} + \hat{e}_i$ верно $\varphi(\tilde{x}) = 1$; 4) $A \cap B = \emptyset$, где $A = \{\tilde{\alpha}\} \cup \{\tilde{x} : \tilde{x} = \tilde{\alpha} + \hat{e}_i, i = 1, 2, \dots, r\}$, $B = \{\tilde{\beta}\} \cup \{\tilde{x} : \tilde{x} = \tilde{\beta} + \hat{e}_i, i = 1, 2, \dots, r\}$.

Множество всех функций $\varphi \in G_0$, а также функций, из которых подстановкой переменных можно получить одну из функций φ , обозначим через G .

Теорема 1. Пусть B — базис, содержащий функцию $\varphi(x_1, x_2, \dots, x_r) \in G$. Тогда для любой булевой функции f существует схема S , реализующая f , для которой $P(S) \leq \varepsilon + 1,1(8(3/2)^r + r(r + 1))\varepsilon^2$ при $\varepsilon \leq \min(1/960, 1/(8(3/2)^r + r(r + 1)))$.

Из теоремы 1 с учётом замечания 1 следует, что при наличии в полном базисе B функций множества G для почти всех булевых функций $f \in K$ асимптотически оптимальные по надёжности схемы S функционируют с ненадёжностью $P(S) \sim \varepsilon$ при $\varepsilon \rightarrow 0$.

ЛИТЕРАТУРА

1. *Лупанов О. П.* Асимптотические оценки сложности управляющих систем. М.: Изд-во Моск. ун-та, 1984.
2. *Васин А. В.* Асимптотически оптимальные по надёжности схемы в полных базисах из трёхходовых элементов: дис. ... канд. физ.-мат. наук. Пенза, 2010. 100 с.

УДК 004.94

КОРРЕКТНОСТЬ ПРАВИЛ ПРЕОБРАЗОВАНИЯ СОСТОЯНИЙ СИСТЕМЫ В РАМКАХ МАНДАТНОЙ СУЩНОСТНО-РОЛЕВОЙ ДП-МОДЕЛИ ОС СЕМЕЙСТВА LINUX

П. Н. Девянин

Рассматриваются де-юре и де-факто правила преобразования состояний системы мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками в операционных системах (ОС) семейства Linux и формулируется утверждение об их корректности относительно заданных в рамках модели требований к реализации мандатного контроля целостности, мандатного и ролевого управления доступом.

Ключевые слова: компьютерная безопасность, формальная модель, управление доступом.

Для строгого формального обоснования безопасности защищённой операционной системы Astra Linux Special Edition [1] автором разрабатывается мандатная сущностно-ролевая ДП-модель управления доступом и информационными потоками в ОС семейства Linux [2, 3] (МРОСЛ ДП-модель). В рамках модели описываются требования к реализации механизмов мандатного контроля целостности, мандатного и ролевого управления доступом, в том числе:

- для уровней доступа и целостности учётной записи пользователя;
- для текущих уровней доступа и целостности субъект-сессии;
- для уровней конфиденциальности и целостности сущности, входящей в состав сущности-контейнера;
- для уровней конфиденциальности и целостности роли или административной роли;
- для уровней конфиденциальности и целостности сущностей, параметрически ассоциированных с учётной записью пользователя, ролью или административной ролью;
- для доступов субъект-сессии к сущности с учётом атрибутов конфиденциальности и целостности сущностей-контейнеров *CCR* и *CCRI*, для создания, удаления сущности или «жёсткой» ссылки на сущность, для сущностей-«дырок», в которых данные «не сохраняются», для доступов субъект-сессии на владение к субъект-сессии, на активизацию из сущности субъект-сессии;
- для специальных сущностей, используемых для получения доступа к сущностям с высоким уровнем целостности;
- для доступов субъект-сессии к роли или административной роли;
- для индивидуальных ролей и административных ролей учётной записи пользователя;
- для изменения атрибутов *CCR*, *CCRI*, переименования роли, административной роли или сущности-контейнера;
- для получения субъект-сессией данных о числе «жёстких» ссылок к сущности;
- для предоставления имён сущностей, ролей или административных ролей;
- для возможности нарушения отдельных условий мандатного управления доступом (при администрировании защищённой ОС).

После этого задаются 20 де-юре и 10 де-факто правил преобразования состояний системы, для каждого из которых детально описываются условия и результаты применения. В таблице приведены примеры задания де-юре правила $grant_rights(x, x')$,

$r, \{(y, \alpha_{rj}): 1 \leq j \leq k\}$), позволяющего субъект-сессии x при кооперации с субъект-сессией x' дать роли r права доступа к сущности y , и де-факто правила $control(x, y, z)$, при реализации которого субъект-сессия x получает фактическое владение субъект-сессией y , используя сущность z , функционально ассоциированную с y .

Примеры задания правил преобразования состояний

Правило	Исходное состояние $G = (PA, user, A, AA, F, H_E)$	Результирующее состояние $G' = (PA', user', A', AA', F', H'_E)$
$grant_rights(x, x', r, \{(y, \alpha_{rj}): 1 \leq j \leq k\})$	$x, x' \in S, y \in E, r \in R \cup AR, (x, r, write_a) \in AA, Constraint_{PA}(PA') = \mathbf{true}, (x, y, own_a) \in A$, [если $y \in S$, то $\alpha_{rj} = own_r$ и $i_s(y) \leq i_r(r)$], [если $y \in E \setminus S$ и $\alpha_{rj} \in \{own_r, write_r\}$, то $i_e(y) \leq i_r(r)$], [если $(y \in S$ и $i_s(y) = i_high)$ или $(y \in E \setminus S$ и $i_e(y) = i_high)$, то $(x', f_s(x)_i_entity, write_a) \in A$], где $1 \leq j \leq k$	$S' = S, E' = E, A' = A, AA' = AA, user' = user, H'_E = H_E, F' = F, PA'(r) = PA(r) \cup \{(y, \alpha_{rj}) : 1 \leq j \leq k\}$ и для $r' \in R \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$
$control(x, y, z)$	$x \in N_S \cap S, y \in S, x \neq y, z \in [y]$ и или $x = z$, или $(x, z, write_m) \in F$, или $[z \in S$ и $z \in de_facto_own(x)]$	$S' = S, E' = E, PA' = PA, A' = A, AA' = AA, user' = user, H'_E = H_E, de_facto_own'(x) = de_facto_own(x) \cup \{y\}, F' = F \cup \{(x, y, write_t), (y, x, write_t)\}$

В результате формулируется и обосновывается утверждение о корректности правил преобразования состояний системы относительно заданных в рамках модели требований к реализации мандатного контроля целостности, мандатного и ролевого управления доступом, т. е. эти требования должны выполняться в состояниях и при переходах между состояниями на всех траекториях функционирования системы.

Утверждение 1. Пусть G_0 — состояние системы $\Sigma(G^*, OP)$, удовлетворяющее требованиям к реализации мандатного контроля целостности, мандатного и ролевого управления доступом. Тогда для любой траектории $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \dots \vdash_{op_N} G_N$, где $N \geq 1$, эти требования выполняются в состоянии G_N и при переходе $G_{N-1} \vdash_{op_N} G_N$.

Таким образом, обеспечивается основа для дальнейшего теоретического исследования и обоснования в рамках МРОСЛ ДП-модели свойств рассматриваемых защищённых ОС семейства Linux.

ЛИТЕРАТУРА

1. Операционные системы Astra Linux // <http://www.astra-linux.ru/>.
2. Десянин П. Н. О разработке мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками в операционных системах семейства Linux // Методы и технические средства обеспечения безопасности информации: Материалы 21-й науч.-технич. конф. 24–29 июня 2012 г. СПб.: Изд-во Политехн. ун-та, 2012. С. 91–94.
3. Десянин П. Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками: учеб. пособие для вузов. 2-е изд., испр. и доп. М.: Горячая линия-Телеком, 2013. 338 с.

УДК 004.435, 004.423, 004.4'41

МОДИФИКАЦИЯ СКОМПИЛИРОВАННЫХ ПРИЛОЖЕНИЙ ДЛЯ ПЛАТФОРМЫ ANDROID МЕТОДОМ АСПЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

Г. Ю. Зайцев, А. И. Потапкин, Д. А. Стефанцов

Описывается инструмент для модификации скомпилированных приложений для платформы Android, разработанный авторами. Инструмент основан на методе аспектно-ориентированного программирования, реализован при помощи библиотеки ASMDEX и объединяет программу приложения и аспекты за два прохода по тексту программы. Аспекты реализуются на языке Java при помощи специальных аннотаций, инкапсулирующих необходимую метаинформацию.

Ключевые слова: АОП, *Android*, *Dalvik*.

Аспектно-ориентированное программирование (АОП) — это способ изменения функциональности программы без изменения исходных текстов последней [1, 2]. Дополнительная функциональность при этом содержится в специальных модулях — аспектах. Удобными для реализации в виде аспектов считаются, например, требования политики безопасности, синхронизация потоков и журналирование [3].

В настоящее время распространены инструменты АОП, предполагающие, что им доступны исходные тексты программы [4, 5]. Однако зачастую требуется изменить функциональность уже скомпилированных программ. В работе описывается разработанный авторами инструмент AspectA (от слов Aspect и Android), позволяющий модифицировать приложения, скомпилированные для платформы Android [6] — одной из самых популярных мобильных платформ на сегодняшний день.

Приложения для платформы Android разрабатываются одним из трёх способов: 1) на языке программирования (ЯП) Java с последующей трансляцией в команды виртуальной машины (ВМ) Dalvik [7]; 2) на любом ЯП, допускающем трансляцию в инструкции для аппаратной архитектуры, на которой работает платформа Android; 3) совмещением двух описанных способов. Распространяются приложения в виде файлов с расширением APK, представляющих собой архив с конфигурационными файлами, файлами с машинными инструкциями для аппаратной архитектуры, файлом с расширением DEX с инструкциями для ВМ Dalvik и другими. Текущая версия AspectA написана на ЯП Java, состоит из трёх компонент (скрипта сборки `inject.sh`, утилиты для объединения DEX-файлов `DexMerger.jar` и утилиты для внедрения вызовов процедур `weaver.jar`) и работает только с приложениями, скомпилированными в инструкции для ВМ.

Интеграция исходного приложения и аспектов осуществляется скриптом `inject.sh` следующим образом: 1) с помощью утилиты `apktool` [8] распаковывается исходный APK-файл; 2) с помощью `DexMerger.jar` DEX-файлы исходного приложения и аспектов объединяются в один; 3) с помощью `weaver.jar` в часть нового DEX-файла, соответствующую исходному приложению, помещаются вызовы процедур из части нового DEX-файла, соответствующей аспектам; 4) новый DEX-файл и старые ресурсы запаковываются в новый APK-файл с помощью утилиты `apktool`. На рис. 1 показан порядок изменения файлов при внедрении аспектов в приложение.

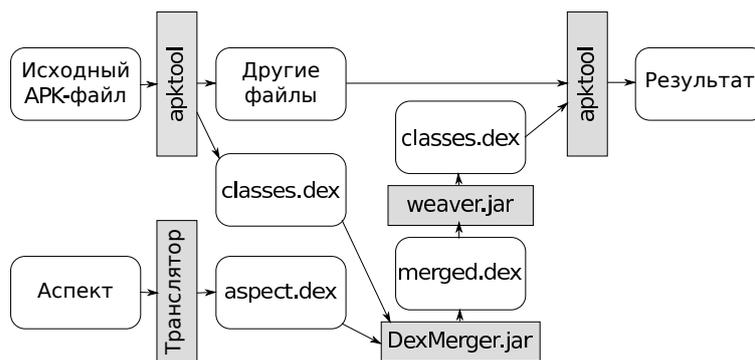


Рис. 1. Общая схема работы AspectA

AspectA реализован с помощью библиотеки ASMDEX [9], позволяющей работать с программой в инструкциях для VM Dalvik. Основную функциональность выполняет компонент `weaver.jar`. Для соединения приложения и аспектов байт-код приложения просматривается в два прохода: во время первого осуществляется поиск областей для внедрения процедур из аспектов, во время второго — указанное внедрение. Во время прохода библиотека ASMDEX просматривает инструкции VM, запуская процедуры-обработчики из `weaver.jar`. Последние анализируют текущую инструкцию и определяют её соответствие описаниям точек выполнения в аспекте. В случае совпадения в анализируемую последовательность инструкций VM помещаются инструкции для вызова предписания из аспекта. Использование двух проходов обусловлено необходимостью сохранения и восстановления регистров VM до и после вызовов внедрённых процедур соответственно. Количество дополнительных регистров, запрашиваемых у VM Dalvik для выполнения этого действия, невозможно вычислить на первом проходе.

Аспекты в описываемом методе должны быть реализованы на языке Java с использованием специальных аннотаций, которые добавляют к конструкциям языка необходимую метаинформацию. Альтернативами к такому подходу могли бы стать адаптация языка AspectJ [4] для работы с байт-кодом Dalvik и разработка собственного аспектно-ориентированного языка. Первый подход слишком трудоёмок по сравнению с использованным решением и, к тому же, обладает всеми ограничениями AspectJ [2]. Второй подход так же трудоёмок и лишает разработчика уже имеющихся инструментов Eclipse IDE [10], работающих для языков Java и AspectJ. В качестве примера на листинге 1 представлен аспект, который осуществляет журналирование всех вызовов член-функций, имена которых начинаются со строки `get`, при этом запись в журнал добавляется перед соответствующим вызовом член-функции (строки 2–8).

```

1 public class TestAspect extends Aspect {
2     public boolean check(JoinPoint jp) {
3         return jp.methodName.startsWith("get");
4     }
5     @BeforeExecution
6     public void advice(JoinPoint jp) {
7         Log.i("invoking method ", jp.methodName);
8     }
9 }
  
```

Листинг 1. Пример аспекта для AspectA

Разработанный инструмент AspectA может применяться для отладки и трассировки скомпилированных приложений для Android во время их разработки и поддержки, при этом разработчики получают возможность легко удалить всю отладочную часть программы. Полезными примерами использования данного инструмента являются также динамический анализ вредоносных приложений и реализация дополнительных механизмов защиты в уже имеющихся приложениях.

ЛИТЕРАТУРА

1. *Filman R. E. and Friedman D. P.* Aspect-oriented programming is quantification and obliviousness [Электронный ресурс] // Technical report, RIACS, 2000. URL: http://www.riacs.edu/research/technical_reports/TR_pdf/TR_01.12.pdf, свободный доступ (дата обращения: 9.04.2010).
2. *Стефанцов Д. А.* Реализация политик безопасности в компьютерных системах с помощью аспектно-ориентированного программирования // Прикладная дискретная математика. 2008. № 1(1). С. 94–100.
3. *Laddad R.* AspectJ in Action: Enterprise AOP with Spring Applications, 2nd edition. Greenwich, CT, USA: Manning Publications Co., 2009. 568 p.
4. <http://eclipse.org/aspectj/> — The AspectJ Project. 2013.
5. <https://sites.google.com/a/gapp.msrg.utoronto.ca/aspectc/> — Welcome to ACC: The AspeCt-oriented C compiler. 2010.
6. <http://www.android.com/about/> — Discover Android. 2013.
7. <http://code.google.com/p/dalvik/> — Dalvik. Code and documentation from Android's VM team. 2011.
8. <https://code.google.com/p/android-apktool/> — Android-Apktool. A tool for reverse engineering Android apk files. 2013.
9. <http://asm.ow2.org/asmdex-index.html> — OW2 Consortium. ASMDEX. 2012.
10. <http://eclipse.org/> — Eclipse. The Eclipse Foundation open source community website. 2013.

УДК 004.94

РАЗРАБОТКА И РЕАЛИЗАЦИЯ МАНДАТНЫХ МЕХАНИЗМОВ УПРАВЛЕНИЯ ДОСТУПОМ В СУБД MYSQL

Д. Н. Колегов, Н. О. Ткаченко, Д. В. Чернов

Работа посвящена разработке и реализации механизмов мандатного управления доступом в изначально дискреционной системе управления базами данных MySQL с использованием формальной модели безопасности. Рассматривается реализация мандатной политики управления доступом типа multilevel security (MLS). Предлагаемый мандатный механизм реализован в составе монитора безопасности ядра MySQL и позволяет выполнить основные требования обеспечения безопасности информационных потоков, предъявляемые к защищённым компьютерным системам. Ключевыми особенностями предлагаемого подхода являются формальное моделирование политики управления доступом на основе аппарата ДП-моделей, реализация мандатного управления доступом на уровне ядра СУБД, а также обеспечение требований безопасности информационных потоков.

Ключевые слова: компьютерная безопасность, управление доступом, информационные потоки, формальные модели безопасности.

В настоящее время в большинстве систем управления базами данных (СУБД) используется дискреционное управление доступом. Вместе с тем некоторые изначально дискреционные СУБД (например, Oracle, Microsoft SQL Server, PostgreSQL) имеют механизмы мандатного управления доступом, которые являются наиболее актуальными и востребованными при построении защищённых компьютерных систем (КС). При этом используемое мандатное управление доступом, как правило, имеет следующие существенные недостатки [1 – 3]:

- отсутствует формальная модель политики управления доступом и информационными потоками;
- не учитываются особенности построения защищённых отечественных КС;
- не учитывается существенное для мандатного управления доступом требование обеспечения безопасности информационных потоков;
- мандатное управление доступом реализовано на прикладном уровне с помощью модификации или перехвата запросов SQL.

С целью исследования вопросов практической реализации мандатного управления доступом в СУБД выполнена разработка и реализация мандатного управления доступом для изначально дискреционной СУБД MySQL. Решены следующие задачи:

- 1) анализ изначально управления доступом;
- 2) разработка формальной модели политики управления доступом и теоретическое обоснование её безопасности;
- 3) реализация механизма мандатного управления доступом на основе построенной модели.

Анализ свойств дискреционного управления доступом СУБД MySQL проводился путём изучения документации, исследования исходного кода и выполнения компьютерных экспериментов. Особое внимание уделено идентификации информационных потоков. Известно, что анализ информационных потоков по времени в большинстве случаев выполнить сложнее, чем анализ информационных потоков по памяти [1]. Было выявлено множество способов реализации информационных потоков по времени (например, информационный поток по времени возникает при блокировании одной субъект-сессией таблицы на запись и попытке записи в неё другой субъект-сессией), но в силу технической сложности корректной реализации механизма мандатного управления доступом в данной работе рассматривается обеспечение безопасности только информационных потоков по памяти.

Установлено, что механизмы реализации информационных потоков по памяти в СУБД MySQL принципиально отличаются от подобных механизмов в ОС. Например, в современных ОС процесс может иметь доступ на чтение и запись к нескольким файлам одновременно. В СУБД MySQL субъект-сессия пользователя, как правило, не может иметь доступ на чтение и запись к двум сущностям СУБД одновременно; исключения составляют механизмы реализации запросов SQL вида

- INSERT INTO ... VALUES((SELECT...), ...);
- INSERT ... SELECT;
- UPDATE ... SET ... = (SELECT ...).

Подобные запросы могут быть использованы для реализации запрещённых информационных потоков по памяти от сущностей с высоким уровнем конфиденциальности к сущностям с низким уровнем конфиденциальности и, таким образом, нарушать требования обеспечения безопасности информационных потоков мандатной политики управления доступом.

На основе проведённого анализа управления доступом в СУБД MySQL сформулированы следующие предположения, используемые в работе:

- 1) информационные потоки рассматриваются только в пределах СУБД;
- 2) рассматриваются только информационные потоки по памяти, порождаемые SQL-операторами SELECT, INSERT, UPDATE и DELETE;
- 3) информационные потоки по времени не рассматриваются.

Для формального моделирования и теоретического обоснования безопасности политик управления доступом и информационными потоками в СУБД MySQL строится промежуточная ДП-модель MySQL, описывающая только мандатное управление доступом типа MLS. Данная модель основана на мандатной ДП-модели [1] и СУБД ДП-модели [4]. В дальнейшем планируется существенно расширить и переработать данную модель, включив в неё элементы политик мандатного управления доступом типа TE и мандатного контроля целостности на основе модели Биба [3].

В настоящей модели используются следующие основные элементы и обозначения:

- O_p — множество сущностей-процедур, O_t — множество сущностей-триггеров, O_v — множество сущностей-представлений, O_{var} — множество сущностей-переменных, O_{gvar} — множество сущностей-глобальных переменных, O_c — множество сущностей-курсоров, COL — множество сущностей-столбцов, O — множество сущностей-объектов, являющееся объединением множеств всех видов, при этом множества сущностей разных видов не пересекаются;
- DB — множество контейнеров-баз данных, TAB — множество контейнеров-таблиц, c_0 — корневой контейнер, содержащий все базы данных и $C = DB \cup TAB \cup \{c_0\}$ — множество сущностей-контейнеров, при этом множества контейнеров не пересекаются друг с другом и с множеством сущностей-объектов;
- S — множество субъект-сессий пользователей, U — множество учётных записей пользователей, при этом $S \cap O = \emptyset$ и $S \cap C = \emptyset$, $E = O \cup C \cup S \cup U$ — множество сущностей;
- (L, \leq) — решётка упорядоченных уровней доступа и уровней конфиденциальности;
- $f_e : (O \setminus O_v) \cup C \rightarrow L$ и $f_s : U \rightarrow L$ — функции, определяющие уровни конфиденциальности и доступа соответственно;
- $R_r = \{alter_r, drop_r, read_r, write_r, append_r, delete_r, execute_r, create_routine_r, create_r, create_user_r, create_trigger_r, create_view_r\}$ — множество прав доступа, $R_a = \{read_a, write_a, append_a\}$ — множество видов доступа, $R_f = \{write_m\}$ — множество информационных потоков;
- $R \subseteq U \times (C \cup O) \times R_r$, $A \subseteq S \times (O \cup C) \times R_a$ — множество текущих прав доступа, $A \subseteq S \times (O \cup C) \times R_a$ — множество текущих доступов, $F \subseteq (E \setminus U) \times (E \setminus U) \times R_f$ — множество текущих информационных потоков.

Функция иерархии сущностей $H : C \cup O_p \cup O_t \cup S \rightarrow 2^{O \cup C}$ отражает то, что любая сущность является либо корневой, либо иерархически подчинённой, а также то, что для любой сущности $e \in E$ существует единственная последовательность сущностей, начинающаяся с контейнера c_0 и заканчивающаяся сущностью $H(e)$ так, что каждый её элемент содержит предыдущий.

Сущность $e \in E$ называется иерархически подчинённой контейнеру $c \in C$, если $e \in H(c)$ или существует $c_2 \in C$, что $e \in H(c_2)$, $c_2 \in H(c)$. Иерархическая подчинённость обозначается знаком $<$.

Введены следующие функции, описывающие элементы мандатного управления доступом типа MLS:

- $user : S \rightarrow U$ – функция, определяющая для каждого субъекта учётную запись пользователя, от имени которой она выполняется;
- $owner : O_p \cup O_t \cup O_v \rightarrow U$ – функция, определяющая для сущностей-процедур, сущностей-триггеров и сущностей-представлений учётные записи, от имени которых они созданы;
- $execute_as : O_p \cup O_t \cup O_v \rightarrow \{as_owner, as_caller\}$ – функция, задающая режим выполнения для процедур, триггеров и представлений и определяющая, от имени какой учётной записи будут выполняться их последовательности правил преобразования;
- $triggers : TAB \times \{write, append, delete\} \rightarrow O_t^*$ – функция, определяющая последовательность триггеров, связанных с таблицей и соответствующим методом обработки;
- $var : S_s \cup O_p \cup O_t \rightarrow 2^{O_v}$ – функция, задающая множество переменных, соответствующих субъект-сессии, процедуре или триггеру;
- $operations : O_p \cup O_t \rightarrow OP^*$ – функция, определяющая последовательность правил преобразования для процедуры или триггера;
- $HLS : O \times C \rightarrow \{\mathbf{true}, \mathbf{false}\}$ – функция, задающая наследование уровней конфиденциальности при доступе к сущностям; $HLS(e, c) = \mathbf{true}$, если $e < c$ или $e = c$, определено значение $f_e(c)$ и не существует $c_1 \in C$, такого, что $e < c_1 < c$ с определённым значением $f_e(c_1)$.

В модели считается, что если пользователь обладает правом доступа на контейнер, то он обладает этим правом доступа также на сущности этого контейнера. При этом предполагается, что если для некоторого $e \in E$ определено значение функции $f_e(e)$, то оно определено и для любого $c \in C$, такого, что $e < c$.

Для описания множества прав доступа на сущности, которые могут быть переданы в рамках сессии пользователя, введено отношение $Grant \subseteq U \times (C \cup O) \times R_r$.

Заданы правила преобразования состояний, описывающие переход СУБД из одного состояния в другое. Существенно новыми являются правила $create_session$, $create_view$, $access_read$, $pass$, $access_append$, $create_user$, $grant_right$, $create_trigger$, $access_write$, $execute_trigger$. В таблице приведены примеры задания некоторых из них.

Примеры правил преобразования состояний

Правило	Исходное состояние	Результирующее состояние
$create_session(u, s)$	$u \in U, s \notin S$	$S'_s = S_s \cup \{s\}, user'(s) = u,$ $f_s(s)' = f_s(u)$
$create_user(s, u, l)$	$s \in S, user(s) \in L_u, u \notin U,$ $l \leq f_s(user(s)),$ $(user(s), c_0, create_user_r) \in R$	$U' = U \cup \{u\}, f'_s(u) = l$
$grant_right(s, u, e, \alpha,$ $grant_option)$	$s \in S, u \in U, e \in C \cup O,$ $\alpha \in R_r, grant_option \in \{\mathbf{true},$ $\mathbf{false}\}, \exists c' \geq e ((s, c', \alpha) \in R_r),$ $\exists c \geq e ((user(s), c, \alpha) \in Grant)$	$R' = R \cup \{(u, e, \alpha)\},$ если $grant_option = \mathbf{true}$, то $Grant' = Grant \cup \{(u, e, \alpha)\}$
$access_read(s, e)$	$s \in S, e \in DB \cup TAB \cup COL,$ $\exists c \in C \cup O (e < c \text{ или } e = c,$ $f_s(user(s)) \geq f_e(c) \text{ и } HLS(e, c) =$ $= \mathbf{true}), \nexists e_1 \in OUC (f_e(e_1) < f_e(e)$ и $(s, e_1, \alpha) \in A$), где $\alpha \in \{write_a,$ $append_a\}$	$A' = A \cup \{(s, e, read_a)\},$ $F' = F \cup \{(e, s, write_m)\}$

Требования политики мандатного управления доступом MLS формулируются через определение ss- и *-свойств состояния модели. В состоянии G системы $\Sigma(G^*, OP)$ доступ $(s, e, \alpha) \in A$ обладает ss-свойством, когда $\alpha = append_a$ или $f_s(user(s)) \leq f_e(e)$. В состоянии G системы $\Sigma(G^*, OP)$ доступы $(s, e_1, read_a), (s, e_2, \alpha) \in A$, где $\alpha \in \{write_a, append_a\}$, обладают *-свойством, если $f_e(e_1) \leq f_e(e_2)$.

В рамках построенной ДП-модели MySQL доказано следующее утверждение, аналогичное базовой теореме безопасности модели Белла — ЛаПадулы [1].

Теорема 1. Пусть начальное состояние G_0 системы $\Sigma(G^*, OP, G_0)$ является безопасным в смысле Белла — ЛаПадулы и $A_0 = F_0 = \emptyset$. Тогда система $\Sigma(G^*, OP, G_0)$ безопасна в смысле Белла — ЛаПадулы.

Механизм мандатного управления доступом реализован на базе MySQL версии 5.5.16 с использованием разработанной ДП-модели. Принятие решения о разрешении или запрете доступа субъект-сессии пользователя к сущности принимается мандатным механизмом после проверки соответствующих прав доступа штатным (дискреционным) механизмом управления доступом. При этом мандатный механизм выполняет

- проверку типа операции (чтение, запись, добавление, удаление) и соответствующего ей вида доступа; уровня доступа учётной записи пользователя и уровня конфиденциальности сущности, что обеспечивает выполнение требований ss-свойства ДП-модели MySQL;
- проверку невозможности реализации информационного потока «сверху вниз» в рамках сессии пользователя, что обеспечивает выполнение требования *-свойства ДП-модели MySQL.

Рассмотрим порядок функционирования мандатного механизма управления доступом, реализующего политику MLS в рамках СУБД MySQL.

Метки безопасности, назначаемые сущностям, хранятся в служебной базе данных *mysql_mac*. Метка безопасности может принимать значение от 0 до 255. Все метки загружаются и хранятся в оперативной памяти через объекты классов *MAC_LABEL*, *MAC_DB*, *MAC_TABLE*, *MAC_COLUMN*, *MAC_EVENT*, *MAC_ROUTINE*, создаваемых на основе данных из *mysql_mac*, что позволяет сохранить уровень производительности первоначальной СУБД. Для загрузки меток безопасности учётных записей пользователей используется уже существующий класс *ACL_USER*. При этом и в том и в другом случае для чтения меток из служебной базы данных *mysql_mac* используются собственные функции MySQL, а для назначения меток — добавленная функция *mac_reload()*. Мандатное управление доступом реализуется следующими функциями:

- *mac_find_type()* — определяет вид доступа по запрашиваемым субъектом-сессией у дискреционного монитора безопасности правам доступа к сущности;
- *mac_ilst_label()* — определяет метки безопасности сущностей, содержащихся в дереве запроса;
- *mac_find_max_label()* — вычисляет максимальное значение среди меток безопасности сущности и содержащих её контейнеров;
- *mac_check_access_ssp()* — обеспечивает выполнение требований ss- и *-свойств.

Рассмотрим порядок выполнения и механизмы функционирования основных функций мандатного управления доступом.

1) После разбора SQL-запроса вызывается собственная функция дискреционного управления доступом *check_access()*, которая производит проверку прав доступа субъект-сессии к сущности, участвующей в этом запросе.

2) Вызываются функции *mac_find_max_label()*, *mac_ilst_label()* и *mac_find_type()*, определяющие необходимые метки безопасности.

3) После получения меток происходит их передача вместе с идентификатором субъект-сессии в функцию *mac_check_access_ssp()*. Первым действием после этого будет конвертация переданных меток в тип *unsigned integer* с целью их последующего корректного сравнения. В зависимости от вида запрашиваемого доступа данной функцией будут выполнены различные проверки. Сначала выполняется проверка условий *ss*-свойства. После того как все условия проверены, инициализируется переменная *current_sec_label*, находящаяся в классе *security_context* и служащая меткой сущности, к которой уже реализован доступ на запись. Далее проверяются условия *-свойства.

4) Проверка возможности реализации доступа субъект-сессии к сущностям осуществляется последовательно. Например, в запросе вида INSERT INTO ... VALUES ((SELECT ...), ...) сначала проверяется возможность реализации доступа вида *append_a* для оператора INSERT, а затем доступа вида *read_a* для каждого оператора SELECT. Таким образом, после первой проверки осуществляется запись в переменную *current_sec_label*, что означает возможность субъект-сессии осуществить запись в сущность, метка безопасности которой равна *current_sec_label*. При следующем вызове этой функции происходит проверка условий *-свойства с использованием метки безопасности сущности и значения переменной *current_sec_label*.

5) В случае предоставления субъект-сессии права доступа на выполнение запросов от имени другой субъект-сессии первая из них будет использовать метки безопасности второй субъект-сессии на время выполнения запросов.

Таким образом, предложена формальная модель безопасности логического управления доступом и информационными потоками (ДП-модель) для СУБД MySQL. Данная модель дополнена элементами мандатного управления доступом, обеспечивающими безопасность информационных потоков. На основе разработанной формальной модели реализован мандатный механизм управления доступом на уровне монитора безопасности ядра СУБД MySQL. Данное решение может быть использовано для построения защищённых СУБД и КС с высоким уровнем доверия к их безопасности. На основе данной работы планируется разработать формальную модель и мандатные механизмы защиты, реализующие другие мандатные политики (например, *type enforcement*, АВАС), учитывающие информационные потоки по памяти для всех операторов SQL, а также основные информационные потоки по времени.

ЛИТЕРАТУРА

1. Десянин П. Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками: учеб. пособие для вузов. М.: Горячая линия-Телеком, 2011. 320 с.
2. Смирнов С. Н. Безопасность систем баз данных. М.: Гелиос АРВ, 2007. 352 с.
3. Bishop M. Computer Security: art and science. Addison-Wesley Professional, 2002. 1084 p.
4. Смольянинов В. Ю. Правила преобразования состояний СУБД ДП-модели // Прикладная дискретная математика. 2013. № 1. С. 50–68.

УДК 004.75: 004.492.3

РАЗРАБОТКА СИСТЕМЫ ОБНАРУЖЕНИЯ РАСПРЕДЕЛЁННЫХ СЕТЕВЫХ АТАК ТИПА «ОТКАЗ В ОБСЛУЖИВАНИИ»

Е. В. Щерба, Д. А. Волков

Предложена методика, разработана архитектура и построена реализация системы обнаружения сетевых атак типа «отказ в обслуживании». Методика основана на моделировании исследуемой сети сетями массового обслуживания с последующей оценкой вероятности потерь заявок в сети.

Ключевые слова: обнаружение сетевых атак, отказ в обслуживании.

Традиционные механизмы обеспечения безопасности — межсетевые экраны и сигнатурные системы обнаружения вторжений — не являются эффективными средствами для обнаружения низкоактивных сетевых атак типа «отказ в обслуживании» (DDoS-атак) прикладного уровня и защиты от них [1].

Фундаментальной предпосылкой для обнаружения атак является построение контрольных характеристик трафика при работе сети в штатных условиях с последующим поиском аномалий в структуре трафика (отклонения от контрольных характеристик). Для обнаружения аномалий могут применяться статистические критерии (среднеквадратичное отклонение, хи-квадрат, отклонение от стандартного нормального распределения, значительное увеличение энтропии и т. д.), кластеризация, метод обнаружения точки перехода, спектральный анализ и др. [2–4]. Каждый из указанных методов имеет определённые достоинства и недостатки и не является универсальным для обнаружения всех типов сетевых атак. Достаточно часто для расчёта параметров потоков данных в вычислительных сетях применяют математические модели в виде сетей массового обслуживания (СеМО). В данной работе для обнаружения DDoS-атак предложен метод оценки вероятности потери произвольной заявки при её прохождении по СеМО. Поскольку атаки прикладного уровня на различные сетевые службы происходят независимо, в рамках каждой службы для моделирования узлов можно использовать одноканальную систему массового обслуживания с очередью длины m .

Рассматривая отдельный узел СеМО, можно предположить, что вся сеть в целом и выбранный узел в частности функционируют в стационарном режиме. Извне поступает пуассоновский поток заявок с параметром λ . Узел содержит одно устройство обслуживания заявок, для которого задана интенсивность μ их обработки. После обработки заявка покидает узел. Если во время поступления заявки обслуживающее устройство занято обработкой другой заявки, то входящая заявка становится в очередь. Если заявка поступает и очередь полностью заполнена, заявка теряется.

Пусть для узлов сети задан вектор интенсивностей входящих потоков заявок извне $\vec{\lambda}$, вектор интенсивностей обработки заявок $\vec{\mu}$ и субстохастическая матрица вероятностей переходов заявок P . В работе [5] предложена итерационная процедура расчёта вектора интенсивностей $\vec{\rho}$ входящих в узлы исходной сети потоков (суммарных потоков извне и из других узлов) и скорректированной субстохастической матрицы вероятностей переходов заявок \tilde{P} . Исходя из потребности оценки вероятности потерь заявок в сети, предложена методика построения цепи Маркова с дискретным временем, соответствующей пути произвольной заявки по узлам. Для этого вводятся расщеплённые состояния этой цепи, т. е. состоянием называется упорядоченная пара чисел (i, d) , где i соответствует номеру узла, в котором находится заявка (меняется в пределах от 1 до J), а d — количеству занятых мест в очереди узла (меняется

в пределах от 1 до $m_i + 1$). Состояние $(i, m_i + 1)$ соответствует переполненной очереди в узле i . Начальное распределение данной цепи \widehat{p} можно рассчитать как

$$\widehat{p}\{(i, d)\} = \frac{\lambda_i \frac{\rho_i^{d-1}}{\mu_i^{d-1}} \left(1 - \frac{\rho_i}{\mu_i}\right)}{\sum_{j=1}^J \lambda_j \frac{\rho_j^{m_j}}{\mu_j^{m_j}}}.$$

Кроме того, вводятся два дополнительных состояния (S) и (F). Первое соответствует успешной обработке заявки, а второе — потере заявки. Начальные вероятности этих состояний равны нулю. Далее определяется матрица вероятностей переходов \widehat{P} заданной цепи Маркова, соответствующей пути произвольной заявки по узлам. Состояния (S) и (F) не сообщаются. Цепь, попав в одно из этих состояний, уже из него не выходит. Вероятность перехода из состояния (i, d) в состояние (j, w) можно рассчитать по формуле

$$p\{(i, d) \rightarrow j, w\} = \widetilde{p}_{ij} \frac{\frac{\rho_j^{w-1}}{\mu_j^{w-1}} \left(1 - \frac{\rho_j}{\mu_j}\right)}{1 - \frac{\rho_j^{m_j}}{\mu_j^{m_j}}},$$

где \widetilde{p}_{ij} — элементы скорректированной субстохастической матрицы вероятностей переходов заявок \widetilde{P} (коррекция необходима, поскольку матрица P устанавливается для СеМО без потерь заявок). Вероятность успешной обработки заявки в узле равна

$$p\{(i, d) \rightarrow S\} = \widetilde{p}_i^*,$$

где \widetilde{p}_i^* — вероятность успешной обработки заявки в узле i (после чего заявка покидает сеть); она вычисляется в ходе итерационной процедуры на основе матрицы P и $p_i^* = 1 - \sum_{k=1}^J p_{ik}$. Если заявка находится в переполненной очереди, вероятность её потери (перехода цепи в состояние (F)) равна

$$p\{(i, m_i + 1) \rightarrow F\} = 1.$$

Все остальные вероятности равны нулю.

Для оценки вероятности потери заявки при стационарном режиме работы сети необходимо рассчитать член вектора $\widehat{p}^{(k)}\{(F)\}$, соответствующий состоянию (F) на k -м шаге заданной цепи Маркова, где $\widehat{p}^{(k)} = \widehat{p} \cdot (\widehat{P})^k$.

Установка параметра k происходит с помощью дополнительной итерационной процедуры. По результатам расчёта вычисляется $\widehat{p}_N^{(k)}$ — вероятность того, что на k -м шаге заданной цепи Маркова заявка всё ещё находится в сети, т. е. не потеряна и не обработана полностью:

$$\widehat{p}_N^{(k)} = 1 - \widehat{p}^{(k)}\{(F)\} - \widehat{p}^{(k)}\{(S)\}.$$

Если в результате вычислений $\widehat{p}_N^{(k)}$ превышает некоторую наперёд заданную точность, то k увеличивается и расчёт повторяется до тех пор, пока не будет достигнута заданная точность указанной вероятности.

На основе представленной методики разработана архитектура и построена программная реализация системы обнаружения DDoS-атак (рис. 1). Разработанная методика позволяет получать адекватную оценку частоты потери заявок в сети в случае,

если СеМО находится в стационарном режиме. При возникновении DDoS-атаки узлы СеМО выходят из стационарного режима на некоторое время, после чего устанавливается стационарный режим с другими параметрами. На время перехода между режимами методика неприменима. Так как время перехода между режимами зависит от топологии сети и параметров узлов, оценка эффективности разработанной методики и её сравнительный анализ с другими подходами представляет отдельную задачу.

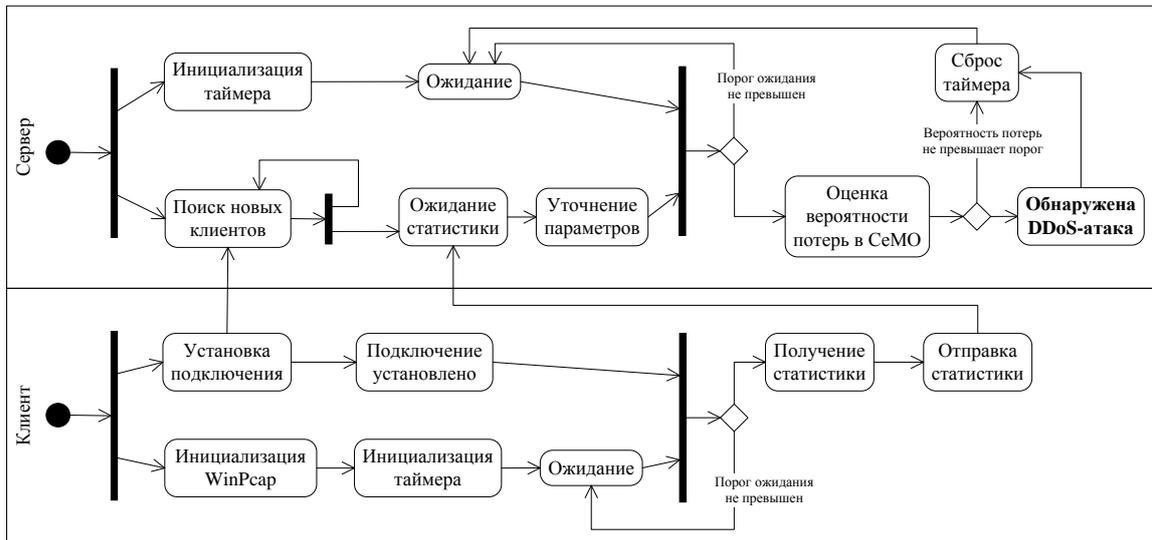


Рис. 1. UML-диаграмма деятельности системы обнаружения

ЛИТЕРАТУРА

1. <http://www.cisco.com/web/RU/netsol/ns480/whitepapers.html> — Предотвращение атак с распределенным отказом в обслуживании (DDoS). Дата обращения: 30.03.2013.
2. *Li Muh., Li Min., and Jiang X.* DDoS attacks detection model and its application // WSEAS Trans. Computers. 2008. V. 7. No. 8. P. 1159–1168.
3. *Wang H., Zhang D., and Shin K. G.* Detecting SYN flooding attacks // Proc. IEEE INFOCOM'2002. New York City, 2002. P. 1530–1539.
4. *Hussain A., Heidemann J., and Papadopoulos C.* A framework for classifying denial of service attacks // Proc. ACM SIGCOMM. Karlsruhe, Germany, 2003. P. 99–110.
5. *Щерба Е. В., Щерба М. В.* Разработка архитектуры системы обнаружения распределенных сетевых атак типа «отказ в обслуживании» // Омский научный вестник. Сер. Приборы, машины и технологии. 2012. № 3 (113). С. 280–283.

Секция 4

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

УДК 519.17

О НИЖНЕЙ ОЦЕНКЕ ЧИСЛА ДОПОЛНИТЕЛЬНЫХ ДУГ
МИНИМАЛЬНОГО ВЕРШИННОГО 1-РАСШИРЕНИЯ
ОРИЕНТАЦИИ ЦЕПИ

М. Б. Абросимов, О. В. Моденова

Даётся нижняя оценка для числа дополнительных дуг минимального вершинного 1-расширения произвольной ориентации цепи.

Ключевые слова: *граф, минимальное вершинное расширение, отказоустойчивость.*

Граф $G^* = (V^*, \alpha^*)$ называется *минимальным вершинным k -расширением n -вершинного графа $G = (V, \alpha)$* , если выполняются следующие условия:

- 1) граф G^* является вершинным k -расширением графа G , то есть граф G вложим в каждый подграф графа G^* , получающийся удалением любых его k вершин;
- 2) граф G^* содержит $n + k$ вершин, то есть $|V^*| = |V| + k$;
- 3) α^* имеет минимальную мощность при выполнении условий 1 и 2.

Понятие минимального вершинного k -расширения появилось в работе Дж. Хейза [1] как модель для исследования отказоустойчивости дискретных систем. Там же доказывается, что минимальным вершинным 1-расширением n -вершинной цепи является $(n + 1)$ -вершинный цикл, а в работе [2] доказывается, что это минимальное вершинное 1-расширение является единственным с точностью до изоморфизма. Задача поиска минимального вершинного k -расширения для произвольного графа является вычислительно сложной [3], и в общем виде решение удалось получить лишь для некоторых классов графов.

Рассмотрим ориентации цепи. Очевидно, что ориентация цепи, являющаяся гамильтоновым графом, имеет единственное с точностью до изоморфизма минимальное вершинное 1-расширение — контур с одной дополнительной вершиной.

В работе [4] доказывается результат, позволяющий связать минимальные вершинные 1-расширения неориентированных графов и их ориентаций: число дополнительных дуг минимального вершинного 1-расширения орграфа не меньше числа дополнительных рёбер минимального вершинного 1-расширения его симметризации.

Удалось установить следующие результаты.

Теорема 1. Среди ориентаций цепи только гамильтонова цепь имеет минимальное вершинное 1-расширение, содержащее две дополнительные дуги.

Теорема 2. Не существует ориентаций цепей с числом вершин $n > 4$, таких, что их минимальное вершинное 1-расширение содержит три дополнительные дуги.

Полученные теоремы позволяют получить следующее утверждение.

Следствие 1. Любая ориентация цепи с числом вершин $n > 4$, отличная от гамильтоновой цепи, имеет минимальное вершинное 1-расширение с не менее чем четырьмя дополнительными дугами.

ЛИТЕРАТУРА

1. Hayes J. P. A graph model for fault-tolerant computing system // IEEE Trans. Comput. 1976. V. C-25. No. 9. P. 875–884.
2. Абросимов М. Б. Графовые модели отказоустойчивости. Саратов: Изд-во Саратов. ун-та, 2012. 192 с.
3. Абросимов М. Б. О сложности некоторых задач, связанных с расширениями графов // Матем. заметки. 2010. Т. 88. Вып. 5. С. 643–650.
4. Абросимов М. Б. Минимальные вершинные расширения направленных звезд // Дискретная математика. 2011. № 23:2. С. 93–102.

УДК 519.172.3, 519.68

СВОЙСТВА ГЕННЫХ СЕТЕЙ ЦИРКУЛЯНТНОГО ТИПА С ПОРОГОВЫМИ ФУНКЦИЯМИ

Ц. Ч.-Д. Батуева

Описан алгоритм нахождения всех неподвижных точек графа состояний генной сети циркулянтного типа с произвольной булевой функцией. Описаны все истоки графа состояний генной сети с пороговой функцией от k переменных, такой, что существует единственный набор v , для которого $f(v) = 1$. Для таких функций от трёх переменных описаны все циклы графа состояний и вычислены длины максимальных цепочек до цикла.

Ключевые слова: генная сеть, ориентированный граф, пороговая функция, граф состояний отображения, цикл, неподвижная точка, исток графа состояний.

Пусть $n \geq k$ — натуральные числа. *Циклическим словом* называется периодическое бесконечное в две стороны слово с периодом n ; обозначается $a_1 a_2 \dots a_n$. Множество всех циклических слов длины n будем обозначать через Ω_n .

Рассмотрим ориентированный граф $G_{n,k+1} = \langle V, E \rangle$, где множество вершин V равно $\{v_1, \dots, v_n\}$ (последовательность вершин соответствует циклическому слову), а множество рёбер E такое, что каждая вершина v_i имеет входящие рёбра из k предыдущих вершин и выходящие в k следующих вершин.

Пороговой функцией называется булева функция, которая представима в виде $f(x_1, \dots, x_k) = \left[\sum_{i=1}^k a_i x_i > T \right]$, где a_i — вес аргумента x_i , а T — порог функции f ; $a_i, T \in \mathbb{R}$.

Рассмотрим пороговую функцию f , зависящую от k переменных. Построим отображение $A_f : \Omega_n \rightarrow \Omega_n$, которое каждому слову $a_1 a_2 \dots a_n$ ставит в соответствие слово $b_1 b_2 \dots b_n$, если

$$b_i = f(a_{i-k}, a_{i-k+1}, \dots, a_{i-1})$$

для всех $i \in \{1, \dots, n\}$. *Неподвижной точкой* отображения A_f называется слово α , такое, что $\alpha = A_f(\alpha)$.

Циклическое слово α называется *истоком* для отображения $A_f : \Omega_n \rightarrow \Omega_n$, если не существует слова β , такого, что $A_f(\beta) = \alpha$.

Графом состояний отображения $A_f : \Omega_n \rightarrow \Omega_n$ называется ориентированный граф, в котором Ω_n — множество вершин, а дуги соединяют слова α и β , если $\beta = A_f(\alpha)$.

Данная работа посвящена описанию свойств графа состояний отображения A_f , а именно описанию истоков, неподвижных точек и циклических состояний, нахождению длин максимальных цепочек.

Пусть f — булева функция от k переменных. Построим ориентированный граф P_f , вершинами которого являются все возможные слова длины k , а рёбра соединяют слова $x_1 \dots x_k$ и $x_2 \dots x_k b$, если $f(x_1, \dots, x_k) = b$. Следующая теорема содержит способ нахождения всех неподвижных точек графов состояния отображения A_f для произвольной булевой функции от k переменных.

Теорема 1. Пусть f — булева функция от k переменных и n кратно l . Граф P_f содержит простой цикл $\langle v_1, \dots, v_l \rangle$, где $v_i = x_i \dots x_{k+i-1}$ для $i \in \{1, \dots, l\}$, тогда и только тогда, когда $(x_1 \dots x_l)^{n/l}$ является неподвижной точкой графа состояний отображения A_f .

В следующей теореме описаны все истоки графа состояний отображения.

Теорема 2. Пусть f — булева функция от k переменных и существует единственный набор значений переменных v , такой, что $f(v) = 1$. Тогда циклическое слово α длины n является истоком графа состояний отображения A_f тогда и только тогда, когда оно удовлетворяет одному из условий:

- 1) α содержит подслово $10^{s-1}1$, где $1 \leq s \leq k-1$ и $s \neq s'$;
- 2) α содержит подслово $10^{k-1}1$, где $k = ts'$ и $t \geq 2$.

Здесь s' — минимальный период слова v .

Получено описание всех циклов и длин максимальных цепочек для графа состояний отображения A_f , где f — булева функция от трёх переменных, удовлетворяющая условию предыдущей теоремы. Если $f(0, 0, 0) \neq 1$, то циклы этого графа представляют циклический сдвиг на 1, 2 или 3 позиции состояния, избегающего определённые слова.

ЛИТЕРАТУРА

1. Евдокимов А. А., Лиховидова Е. О. Дискретная модель генной сети циркулянтного типа с пороговыми функциями // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2008. №2(3). С. 18–21.

УДК 519.17

К ВОПРОСУ О ВЕРХНЕЙ ОЦЕНКЕ ЧИСЛА ДОПОЛНИТЕЛЬНЫХ РЁБЕР МИНИМАЛЬНЫХ ВЕРШИННЫХ РАСШИРЕНИЙ ЦВЕТНЫХ ЦИКЛОВ

П. П. Бондаренко

Приводится верхняя оценка количества дополнительных рёбер в минимальных вершинных 1-расширениях циклов с вершинами двух типов, а также общий вид одного из расширений.

Ключевые слова: граф, цикл, минимальное расширение, отказоустойчивость.

Будем рассматривать неориентированные графы с вершинами двух типов или цветов. Для исследования отказоустойчивости дискретных систем J. P. Hayes [1] предло-

жил графовую модель и рассмотрел отказоустойчивые реализации некоторых классов графов. Основные понятия используются в соответствии с работой [2].

Определение 1. Граф $G^* = (V^*, \alpha^*)$ называется *минимальным вершинным k -расширением n -вершинного графа $G = (V, \alpha)$ с вершинами p типов*, если выполняются следующие условия:

- 1) граф G^* является вершинным k -расширением графа G , то есть граф G вложим в каждый подграф графа G^* , получающийся удалением любых его k вершин;
- 2) граф G^* содержит $n + k \cdot p$ вершин, то есть $|V^*| = |V| + k \cdot p$;
- 3) α^* имеет минимальную мощность при выполнении условий 1 и 2.

Рассмотрим цепь P_n с $n + 2$ вершинами двух типов: двумя вершинами первого типа степени 1 и n вершинами второго типа. Найдём минимальный по количеству рёбер граф P_n^* , состоящий из $n + 3$ вершин (две вершины первого типа, а остальные — второго), такой, что при удалении любой вершины второго типа существует путь между вершинами первого типа, проходящий по n вершинам второго типа.

Лемма 1. Минимальное количество дополнительных рёбер в P_n^* равно $\lfloor n/2 \rfloor + 3$. Если вершины цепи P_n пронумерованы от 0 до $n + 1$, добавленная вершина в цепи P_n^* имеет номер $n + 2$, то для получения P_n^* к P_n нужно добавить следующие рёбра:

- а) n — нечётное, $n > 1$:
 $\{0, 2\}; \{n + 2, n - 2\}; \{n + 2, n\}; \{n + 2, n + 1\}; \{k, k + 3\}, 1 \leq k \leq n - 4$;
- б) n — чётное, $n > 2$:
 $\{0, 2\}; \{n + 2, n - 3\}; \{n + 2, n - 1\}; \{n + 2, n\}; \{n + 2, n + 1\}; \{k, k + 3\}, 1 \leq k \leq n - 5$;
- в) $n = 1$: $\{n + 2, 0\}, \{n + 2, 2\}$;
- г) $n = 2$: $\{n + 2, 0\}, \{n + 2, 1\}, \{n + 2, 2\}, \{n + 2, 3\}$.

На рис. 1 приведены иллюстрирующие лемму примеры.

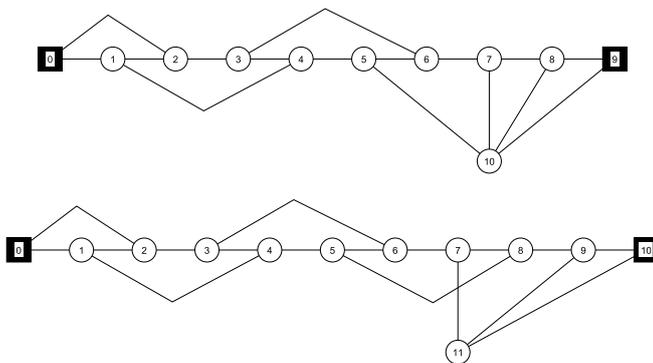


Рис. 1. Примеры P_n^* для $n = 9$ и 10

Будем рассматривать циклы C_n с вершинами двух типов. Поставим каждому циклу в соответствие последовательность чисел e_0, e_1, \dots, e_{k-1} , такую, что

- 1) k — количество рёбер в цикле C_n , соединяющих две вершины разного типа, т. е. количество групп последовательных вершин одного типа, причём вершины, соседние с крайними из каждой группы, имеют другой тип;
- 2) i -я группа содержит e_i последовательно соединённых вершин одного типа. Пусть тип вершин в i -й группе отличается от типа вершин в $(i - 1)$ -й группе для любого $i > 0$. Типы вершин в нулевой и $(k - 1)$ -й группе тоже не совпадают;
- 3) $e_0 + e_1 + \dots + e_{k-1} = n$.

Такой цикл будем обозначать $C_n(e_0, e_1, \dots, e_{k-1})$.

Пусть C_n^* — минимальное вершинное 1-расширение цикла $C_n(e_0, e_1, \dots, e_{k-1})$. Тогда минимальное количество рёбер, которые нужно добавить к C_n , чтобы получить C_n^* , можно оценить следующим образом.

Теорема 1. Количество дополнительных рёбер m в минимальном вершинном 1-расширении цикла $C_n(e_0, e_1, \dots, e_{k-1})$ C_n^* удовлетворяет условию

$$m < \sum_{i=0}^{k-1} \lfloor e_i/2 \rfloor + 3 \leq \lfloor n/2 \rfloor + 3k.$$

Одно из вершинных 1-расширений строится по лемме 1 последовательным рассмотрением всех групп e_i .

ЛИТЕРАТУРА

1. Hayes J. P. A graph model for fault-tolerant computing system // IEEE Trans. Comput. 1976. V. C-25. No. 9. P. 875–884.
2. Абросимов М. Б. Графовые модели отказоустойчивости. Саратов : Изд-во Сарат. ун-та, 2012.

УДК 519.7

ИССЛЕДОВАНИЕ ДИНАМИЧЕСКИХ СВОЙСТВ НЕКОТОРЫХ ДИСКРЕТНО-АВТОМАТНЫХ ОТОБРАЖЕНИЙ, ЗАДАННЫХ СЛУЧАЙНЫМИ ГРАФАМИ¹

А. А. Евдокимов, С. Е. Кочемазов, И. В. Отпущенников, А. А. Семенов

Приведены результаты вычислительного анализа задач поиска неподвижных точек и циклических режимов (циклов) для ряда дискретных отображений, используемых при моделировании поведения систем со множеством взаимодействующих агентов. Рассматривались отображения, задаваемые случайными графами, сгенерированными в соответствии с известными моделями (G_{np} -графы, модель Уоттса — Строгатца).

Ключевые слова: случайные графы, генные сети, дискретно-автоматные отображения, SAT.

В последние несколько лет набирают популярность задачи исследования различных свойств мультиагентных систем, взаимодействие компонентов которых определяется сетями [1,2]. Такие системы используются в биоинформатике [3], в исследовании информационных и социальных сетей [2], а также в экономическом моделировании [4]. Авторами в течение ряда лет рассматривались задачи исследования динамических свойств дискретных отображений, естественным образом связанных с сетями. Так, в [5] введены и исследованы дискретные модели, описывающие процессы в генных сетях, получены теоретические результаты (в форме теорем), дающие условия возникновения неподвижных точек и циклов для отображений, заданных сетью регулярной структуры (использовались циркулянтные графы). В работе [6] весовые функции из [5] использовались в сетях случайной структуры. Рассматривались задачи поиска неподвижных

¹Работа выполнена при поддержке Междисциплинарного интеграционного проекта СО РАН № 80 «Дифференциально-разностные и интегродифференциальные уравнения. Приложения к задачам естествознания»; гранта Президента РФ СП-3667.2013.5; грантов РФФИ № 11-07-00377а, 11-01-00997.

точек и циклов для соответствующих дискретно-автоматных отображений. Для численного решения этих задач применён SAT-подход [7]. Удалось успешно решить задачи поиска неподвижных точек для отображений, заданных сетями с несколькими сотнями вершин. В работе [8] предложена общая дискретная модель генных сетей с учетом различных регуляторных факторов агентов, таких, как активация, ингибирование и авторегуляция.

В настоящей работе представлены новые результаты по исследованию динамических свойств дискретно-автоматных отображений, задаваемых сетями, сгенерированными в соответствии с известными моделями случайных графов (G_{np} -модель, модель Уоттса — Строгатца [2]). В рассматриваемых сетях использованы весовые функции узлов, предложенные в [8]. Для поиска стационарных состояний (неподвижных точек) и циклических режимов (циклов) применен SAT-подход [7]. Для сетей с десятками вершин за разумное время удалось найти большое число неподвижных точек. Экспериментально показано, что наличие циклов малой длины для рассматриваемых отображений находится в обратной зависимости от разреженности графа сети (чем разреженнее граф, тем меньше шансов существования циклов).

ЛИТЕРАТУРА

1. Newman M. E. J. The structure and function of complex networks // SIAM Review. 2003. V. 45. P. 167–256.
2. Dorogovtsev S. N., Goltsev A. V., and Mendes J. F. F. Critical phenomena in complex networks // Rev. Mod. Phys. 2008. V. 80. P. 1275–1335.
3. Системная компьютерная биология / под ред. Н. А. Колчанова, В. А. Гончарова, В. А. Лихошвая, В. А. Иванисенко. Новосибирск: Изд-во СО РАН, 2008.
4. Vitali S., Glattfelder J., and Battiston S. The network of global corporate control // PLoS ONE 6(10): e25995. doi: 10.1371/journal.pone.0025995.
5. Григоренко Е. Д., Евдокимов А. А., Лихошвай В. А., Лобарева И. А. Неподвижные точки и циклы автоматных отображений, моделирующих функционирование генных сетей // Вестник Томского государственного университета. Приложение. 2005. № 14. С. 206–212.
6. Евдокимов А. А., Кочемазов С. Е., Семенов А. А. Применение символьных вычислений к исследованию дискретных моделей некоторых классов генных сетей // Вычислительные технологии. 2011. Т. 16. № 1. С. 30–47.
7. Biere A., Heule V., van Maaren H., and Walsh T. Handbook of Satisfiability. IOS Press, 2009.
8. Евдокимов А. А., Кочемазов С. Е., Отпущенников И. В., Семенов А. А. Символьные алгоритмы решения булевых уравнений в применении к исследованию дискретных моделей генных сетей // Материалы XVI Междунар. конф. «Проблемы теоретической кибернетики». Н. Новгород, 2011. С. 151–154.

УДК 519.1

О ВЕТВЛЕНИИ И НЕПОСРЕДСТВЕННЫХ ПРЕДШЕСТВЕННИКАХ СОСТОЯНИЙ В КОНЕЧНОЙ ДИНАМИЧЕСКОЙ СИСТЕМЕ ВСЕХ ВОЗМОЖНЫХ ОРИЕНТАЦИЙ ГРАФА

А. В. Жаркова

Подсчитывается ветвление и определяются непосредственные предшественники состояний в конечной динамической системе, состояниями которой являются все возможные ориентации данного графа, а эволюционная функция задаётся следующим образом: динамическим образом данного орграфа является орграф, полу-

ченный из исходного путём переориентации всех дуг, входящих в стоки, других отличий между исходным орграфом и его образом нет. Определяется также свойство недостижимости состояния в данной динамической системе.

Ключевые слова: *конечная динамическая система, граф, ориентация графа, ветвление, недостижимость, непосредственный предшественник.*

Под *конечной динамической системой* понимается пара (S, δ) , где S — конечное непустое множество, элементы которого называются состояниями системы; $\delta: S \rightarrow S$ — отображение множества состояний в себя, называемое *эволюционной функцией системы*. Таким образом, каждой конечной динамической системе сопоставляется карта, представляющая собой орграф с множеством вершин S и дугами, проведенными из каждой вершины $s \in S$ в вершину $\delta(s)$. Компоненты связности графа, задающего динамическую систему, называются её *бассейнами*. Получается, что каждый бассейн представляет собой контур с входящими в него деревьями. Контур, в свою очередь, называется предельными циклами, или *аттракторами*.

К числу основных характеристик состояний динамических систем можно отнести *ветвление* (количество непосредственных предшественников данного состояния), а также свойство недостижимости состояния (то есть когда состояние имеет нулевое ветвление; состояния, обладающие данным свойством, называются также *начальными состояниями системы*). Автором написаны программы для ЭВМ, позволяющие вычислять различные параметры динамических систем, ассоциированных с некоторыми типами графов [1].

Пусть дан некоторый граф G . Придадим его рёбрам произвольную ориентацию, тем самым получив ориентированный граф \vec{G} . Применим к полученному орграфу эволюционную функцию α , которая у данного орграфа одновременно переориентирует все дуги, входящие в стоки (вершины с нулевой степенью исхода), а остальные дуги оставляет без изменения, в результате чего получаем орграф $\alpha(\vec{G})$. Если проделать данные действия со всеми возможными ориентациями данного графа, то получим карту динамической системы заданной размерности (что определяется количеством рёбер в графе), состоящую из одного или нескольких бассейнов. Данная динамика для бесконтурных связных орграфов введена в [2]. Итак, будем рассматривать динамическую систему (Γ_G, α) , где через Γ_G обозначим множество всех возможных ориентаций данного графа G , а эволюционная функция α задаётся следующим образом: если дан некоторый орграф $\vec{G} \in \Gamma_G$, то его динамическим образом $\alpha(\vec{G})$ является орграф, полученный из \vec{G} одновременной переориентацией всех дуг, входящих в стоки, других отличий между \vec{G} и $\alpha(\vec{G})$ нет.

Определим ветвление состояний, а также их непосредственных предшественников в полученной системе. Автором, в частности, рассматривались ветвления и непосредственные предшественники состояний в конечных динамических системах, ассоциированных с некоторыми типами графов, например цепями [3].

В [2] состояниями системы являются бесконтурные связные орграфы и замечается, что для любого достижимого состояния s рассматриваемой системы и состояния $s' = \alpha(s)$ каждый сток в s' имеет по крайней мере одну смежную вершину, которая также является стоком и в s . Ставится вопрос об определении всех возможных непосредственных предшественников состояния s' данной динамической системы. Из определённого свойства автор книги замечает, что каждый сток в s' имеет по крайней мере одну смежную вершину, являющуюся источником (вершиной с нулевой степенью захода), и тогда ответ на вопрос может быть получен путём построения всех таких непу-

стых подмножеств множества источников в орграфе, представляющем состояние s' , которые смежны со всеми стоками. Тогда количество непосредственных предшественников данного состояния s' равно количеству таких подмножеств; если же для данной ориентации графа таких подмножеств не существует, то такое состояние является начальным.

Определение 1. Множество источников ориентированного графа назовем *допустимым*, если из него в каждый сток этого графа есть дуга.

Теорема 1. Ветвление данного состояния s динамической системы (Γ_G, α) равно:

- 1) количеству допустимых множеств источников в орграфе \vec{G} , представляющем состояние s , если в \vec{G} есть стоки;
- 2) количеству различных подмножеств множества источников, включая пустое, в орграфе \vec{G} , представляющем состояние s , если в \vec{G} нет стоков.

Следствие 1. Состояние s динамической системы (Γ_G, α) недостижимо тогда и только тогда, когда в орграфе \vec{G} , представляющем состояние s , есть по крайней мере один сток и при этом нет ни одного допустимого множества источников, или, другими словами, когда существует хотя бы один сток в \vec{G} , не смежный с источниками.

Следствие 2. Для нена начального состояния s динамической системы (Γ_G, α) все его непосредственные предшественники определяются:

- 1) всеми различными допустимыми множествами источников в орграфе \vec{G} , представляющем состояние s , если в \vec{G} есть стоки;
- 2) всеми подмножествами множества источников, включая пустое, в орграфе \vec{G} , представляющем состояние s , если в \vec{G} нет стоков.

Это определение происходит следующим образом: все дуги, исходящие из всех источников соответствующего множества, переориентируются, а все остальные дуги остаются без изменения.

ЛИТЕРАТУРА

1. Власова А. В. Исследование эволюционных параметров в динамических системах двоичных векторов // Свидетельство о государственной регистрации программы для ЭВМ № 2009614409, выданное Роспатентом. Заявка № 2009613140. Дата поступления 22 июня 2009 г. Зарегистрировано в Реестре программ для ЭВМ 20 августа 2009 г.
2. Barbosa V. C. An atlas of edge-reversal dynamics. London: Chapman&Hall/CRC, 2001.
3. Власова А. В. Об одной динамической системе / Саратов. гос. ун-т. Саратов, 2007. 17 с. Библиогр.: 2 назв. Рус. Деп. в ВИНТИ 17.12.07, № 1181-B2007.

УДК 519.17

О МИНИМАЛЬНЫХ РЁБЕРНЫХ РАСШИРЕНИЯХ ПАЛЬМ СПЕЦИАЛЬНОГО ВИДА

Д. Д. Комаров

Описаны минимальные рёберные 1-расширения графов специального класса — двулистных пальм.

Ключевые слова: расширения графов, деревья, пальмы.

Ф. Харари и Дж. Хейз в работе [1] рассматривают граф как модель некоторой технической системы. Вершины графа — её элементы, а ребра — связи между ними. Отказ связи системы рассматривается как удаление соответствующего этой связи ребра. При такой интерпретации минимальное рёберное k -расширение графа, моделирующего некоторую систему Σ , является моделью оптимальной рёберной k -отказоустойчивой реализации системы Σ .

Задача нахождения минимального рёберного расширения произвольного графа является NP-полной [2], поэтому представляет интерес нахождение классов графов, для которых можно построить минимальное рёберное расширение аналитически.

Определение 1. Назовём граф G^* рёберным k -расширением графа G , если G вкладывается в каждый граф, получающийся из G^* удалением любых его k ребер.

Определение 2. Граф $G^* = (V^*, \alpha^*)$ называется минимальным рёберным k -расширением графа $G = (V, \alpha)$, если выполняются следующие условия:

- 1) G^* является рёберным k -расширением G ;
- 2) $|V^*| = |V|$;
- 3) α^* имеет минимальную мощность при выполнении условий 1 и 2.

Определение 3. *Сверхстройным деревом* называется корневое дерево, где степень всех вершин, кроме корня, не превосходит 2, а степень корня более 2.

Альтернативное определение:

Граф G называется сверхстройным деревом, если он является объединением $s > 2$ цепей P_1, \dots, P_s с общей концевой вершиной.

Определение 4. Назовем сверхстройное дерево r -листной пальмой высоты r , если оно образовано объединением $s > 2$ цепей P_1, \dots, P_s с общей концевой вершиной, причём длина цепи P_1 равна r , а длины остальных цепей равны 1.

Определение 5. Назовем *рогатым циклом* длины n с k рогами граф, полученный из n -звенного цикла и k трёхзвенных циклов таким образом, что каждый из трёхзвенных циклов имеет ровно одно общее ребро с n -звенным циклом и ни один из трёхзвенных циклов не имеет общих рёбер с другими трёхзвенными циклами. При этом назовём n -звенный цикл *телом* рогатого цикла, а трёхзвенные циклы — его *рогами*.

Определение 6. Назовём *разреженностью* рогатого цикла длину максимального пути между вершинами, принадлежащими разным рогами, проходящего по рёбрам, не принадлежащим ни одному из рогов.

На рис. 1 показаны рогатый цикл длины 6 с тремя рогами и разреженностью 2 (а) и рогатый цикл длины 8 с двумя рогами и разреженностью 3 (б).

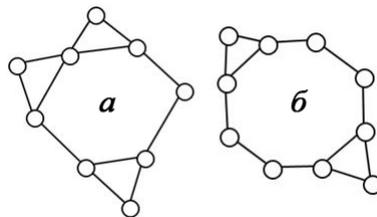


Рис. 1. Примеры рогатых циклов

Теорема 1. Пусть граф G — двулистная пальма высоты n , $n > 3$. Тогда рогатый цикл G_1 с количеством рогов $p = \left\lceil \frac{n-4}{6} \right\rceil + 2$, длиной $n_1 = n - p + 3$ и разреженностью меньше 5 является минимальным рёберным 1-расширением графа G .

ЛИТЕРАТУРА

1. Harary F. and Hayes J. P. Edge fault tolerance in graphs // Networks. 1993. V. 23. P. 135–142.
2. Абросимов М. Б. О сложности некоторых задач, связанных с расширениями графов // Матем. заметки. 2010. Т. 88. № 5. С. 643–650.

УДК 519.174

ДЕРЕВЬЯ ФУНКЦИОНАЛЬНЫХ ГРАФОВ ДЛЯ ЦИРКУЛЯНТОВ С ЛИНЕЙНЫМИ БУЛЕВЫМИ ФУНКЦИЯМИ В ВЕРШИНАХ

А. С. Корниенко

Получено описание функционального графа дискретной динамической системы, являющейся моделью регуляторного контура геной сети.

Ключевые слова: дискретная динамическая система, циркулянт, геной сеть, регуляторный контур, функциональный граф.

Пусть даны $n \geq 3$, $\{d_1, d_2, \dots, d_k\} \subseteq \{0, 1, \dots, n-1\}$ и ориентированный граф $G_{n;d_1, d_2, \dots, d_k}$ с множествами вершин $\{0, 1, \dots, n-1\}$ и дуг $\{\vec{ij} : (j-i) \equiv d_r \pmod{n}, r = 1, 2, \dots, k\}$. Матрица смежности таких графов называется циркулянтом. Эти графы также принято называть циркулянтами [1].

Рассмотрим следующую дискретную динамическую систему. В каждый момент времени вершины циркулянта $G_{n;d_1, d_2, \dots, d_k}$ помечены элементами v_0, v_1, \dots, v_{n-1} из конечного поля F_q порядка q . Набор $\tilde{v} = (v_0, v_1, \dots, v_{n-1}) \in F_q^n$ назовём состоянием системы. В следующий момент времени (такт работы системы) состояние системы меняется, и динамика его изменения определяется отображением

$$A_{f,q} : F_q^n \rightarrow F_q^n,$$

где $f = (f_0, f_1, \dots, f_{n-1})$ и новая метка каждой вершины i является значением функции $f_i : F_q^k \rightarrow F_q$, аргументы которой принимают значения старых меток в тех вершинах, дуги из которых входят в вершину i .

Функциональным графом $G_{f,q}$ называется ориентированный граф, вершинами которого являются элементы F_q^n , причём дуга из вершины \tilde{v} идёт в вершину \tilde{u} тогда и только тогда, когда $A_{f,q}(\tilde{v}) = \tilde{u}$.

В работе рассматривается структура функционального графа в случае, когда $q = 2$, все функции f_i равны между собой и линейны и отображение $A_{f,2}$ действует следующим образом:

$$A_{f,2}(v_0, v_1, \dots, v_{n-1}) = (u_0, u_1, \dots, u_{n-1}),$$

$$u_i = v_{i-1} + v_i + v_{i+1}, \quad i = 0, 1, \dots, n-1, \quad \text{где } v_{-1} = v_{n-1}, v_n = v_0.$$

С использованием методов, изложенных в [2], доказаны следующие свойства функционального графа $G_{f,2}$:

- если n не кратно 3, то отображение $A_{f,2}$ обратимо и функциональный граф $G_{f,2}$ является дизъюнктивным объединением простых контуров;

- в функциональном графе $G_{f,2}$ при $n = 3 \cdot 2^k(2m + 1)$ множество вершин, принадлежащих деревьям, разбивается на 2^k уровней, причём каждая вершина дерева имеет ровно четырёх предков;
- при чётном n функциональный граф $G_{f,2}$ содержит четыре неподвижные точки, при нечётном n — две неподвижные точки;
- если $n = 2^k(2m + 1)$, то все длины циклов функционального графа $G_{f,2}$ являются делителями $2^k(2^s - 1)$, где $s = \min\{j : j > 0, 2^j \equiv \pm 1 \pmod{2m + 1}\}$.

ЛИТЕРАТУРА

1. Харари Ф. Теория графов. М.: УРСС, 2003.
2. Евдокимов А. А., Пережогин А. Л. Дискретные динамические системы циркулянтного типа с линейными функциями в вершинах сети // Дискретный анализ и исследование операций. 2011. Т. 3. № 3. С. 39–48.

УДК 519.6

О ЛОКАЛЬНОЙ ПРИМИТИВНОСТИ ГРАФОВ И НЕОТРИЦАТЕЛЬНЫХ МАТРИЦ

С. Н. Кяжин

Положительным криптографическим свойством генератора гаммы, построенного на основе управляющего и генерирующего блоков, является существенная зависимость элементов состояний генерирующего блока от всех знаков начального состояния генератора. Для изучения такого рода зависимостей в рамках матрично-графового подхода введено понятие локальной примитивности неотрицательных матриц и графов. Получены условия локальной примитивности матриц. Установлена связь характеристик локальной примитивности частного класса матриц (графов) с конструктивными параметрами генераторов гаммы.

Ключевые слова: экспонент, локальный экспонент, примитивная матрица, примитивный граф, локальная примитивность.

Пусть $M_0(n)$ — множество всех квадратных неотрицательных матриц порядка n , $A \in M_0(n)$, $J = \{j_1, \dots, j_r\}$, $\emptyset \neq J \subseteq \{1, \dots, n\}$; $A(J^2)$ — подматрица порядка r , полученная из A вычёркиванием строк и столбцов с номерами $j \neq j_1, \dots, j_r$. Множество матриц, для которых подматрица $A(J^2)$ неотрицательна, обозначим $M_0(J^2)$.

Матрицу A назовем J^2 -положительной, если положительна подматрица $A(J^2)$. Обозначим $M_+(J^2)$ полугруппу по умножению J^2 -положительных матриц.

Матрица A называется квазиположительной, если все её строки и столбцы отличны от нулевых. Матрица A называется J^2 -квазиположительной, если квазиположительной является подматрица $A(J^2)$. Обозначим $Q(J^2)$ множество J^2 -квазиположительных матриц.

Квазиположительную матрицу A назовём J^2 -примитивной, если положительна подматрица $A^t(J^2)$ матрицы A^t при любом натуральном $t \geq \gamma$; наименьшее такое число γ назовём J^2 -экспонентом матрицы A и обозначим $J^2\text{-exp}A$. Множество J^2 -примитивных матриц обозначим $P(J^2)$.

Подматрицу размера $n \times r$, полученную из A вычёркиванием столбцов с номерами $j \neq j_1, \dots, j_r$, обозначим $A(J)$ и назовём её в соответствующих условиях J -положительной (J -квазиположительной, J -примитивной). Множества таких матриц обозначим соответственно $M_+(J)$, $Q(J)$ и $P(J)$. Наименьшее натуральное число γ , при кото-

ром подматрицы $A^t(J)$ строго положительны при любом $t \geq \gamma$, назовём J -экспонентом матрицы A и обозначим J -ехр A .

Обозначим $S(J^2)$ группу подстановочных матриц порядка n , для которых любой элемент $i \notin J$ неподвижный.

Утверждение 1. При любом непустом подмножестве $J \subseteq \{1, \dots, n\}$, $n > 1$, имеет место:

- 1) $M_+(J^2) \subset P(J^2) \subset Q(J^2)$;
- 2) $S(J^2) \subset Q(J^2)$, при этом $S(J^2) \cup P(J^2) = \emptyset$;
- 3) множество $Q(J^2)$ образует моноид относительно умножения, $M_+(J^2)$ — идеал моноида $Q(J^2)$.

Пусть $A, B \in M_0(n)$, $A = (a_{ij})$, $B = (b_{ij})$, определим отношение

$$A \leq B \Leftrightarrow a_{ij} \leq b_{ij}, \quad i, j = 1, \dots, n.$$

Утверждение 2.

- 1) Пусть $J \leq I$, тогда если матрица A не J^2 -примитивная, то она не I^2 -примитивная; если матрица A является I^2 -примитивной, то она J^2 -примитивная и J^2 -ехр $A \leq I^2$ -ехр A ; аналогичные соотношения верны для J -примитивности и I -примитивности.
- 2) Если матрица A не J^2 -примитивная, то и матрица $A(J^2)$ не примитивная; если матрица $A(J^2)$ примитивная, то матрица A является J^2 -примитивной и J^2 -ехр $A \leq$ ехр $A(J^2)$.
- 3) Если $A \leq B$ и A является J^2 -примитивной, то и B является J^2 -примитивной и J^2 -ехр $A \leq J^2$ -ехр B .

Утверждение 3. Если матрицы $A, B \in M_0(J^2)$ сопряжены в группе $S(J^2)$, то A и B одновременно или J^2 -примитивны, или не J^2 -примитивны.

Обозначим через $\Gamma(A)$ орграф, матрицей смежности вершин которого является носитель матрицы A . Известно, что матрица A и граф $\Gamma(A)$ одновременно примитивны или не примитивны.

Теорема 1. Граф $\Gamma(A)$ является J^2 -примитивным тогда и только тогда, когда $\Gamma(A)$ имеет сильносвязный примитивный подграф с множеством вершин, содержащим J .

Теорема 2. Связный граф $\Gamma(A)$ является J -примитивным тогда и только тогда, когда $\Gamma(A)$ имеет сильносвязный примитивный подграф U с множеством вершин, содержащим J , и из каждой вершины $i \notin U$ достижимо множество вершин U .

Следствие 1. Пусть J^2 -ехр $A = \gamma$, J -ехр $A = \delta$. Если граф $\Gamma(A)$ является J -примитивным, то $\gamma \leq \delta \leq \gamma + \max_i \rho(i, U)$, где $\rho(i, U)$ — длина кратчайшего пути из вершины $i \notin U$ в ближайшую вершину $j \in U$.

Ряд генераторов гаммы построен на основе последовательного соединения управляющего и генерирующего автоматов [1, гл. 18], при этом выходная гамма образуется с помощью функции от некоторого подмножества J знаков состояний генерирующего автомата. Положительным криптографическим свойством такого генератора является

существенная зависимость всех знаков множества J от всех знаков начального состояния генератора. В связи с этим рассмотрим автономный автомат без выхода A , построенный как последовательное соединение двух регистров правого сдвига длины n и m соответственно с функциями обратной связи $f(x)$ и $g(x)$.

Пусть V_r — множество двоичных r -мерных векторов, $r = 1, 2, \dots$; A — автомат с множеством состояний V_{n+m} , выходным алфавитом управляющего регистра V_1 и функцией переходов h :

$$h(x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}) = \\ = (f(x_1, \dots, x_n), x_1, \dots, x_{n-1}, x_n \oplus g(x_{n+1}, \dots, x_{n+m}), x_{n+1}, \dots, x_{n+m-1}).$$

Перемешивающая матрица M (порядка $m+n$) преобразования h генератора имеет вид

$$M = \begin{pmatrix} A & B \\ 0 & C \end{pmatrix},$$

где A — перемешивающая матрица порядка n преобразования управляющего регистра; C — матрица порядка m , совпадающая при фиксированном x_n с перемешивающей матрицей преобразования генерирующего регистра; в матрице $B = (b_{ij})$ порядка $n \times m$ элемент $b_{n,n+1} = 1$, а остальные элементы равны 0.

Теорема 3. Пусть $J = \{n+1, \dots, n+m\}$, функция $g(x)$ существенно зависит от переменных с номерами j_1, \dots, j_r, m , где $1 \leq j_1 < \dots < j_r < m$, $1 \leq r < m$, $\text{НОД}(j_1, \dots, j_r, m) = d \geq 1$. Тогда матрица M преобразования генератора J -примитивна, если и только если $d = 1$; в случае J -примитивности верны следующие оценки:

- 1) $J\text{-exp } M \leq \max\{n + j_1(m-1), \text{exp } C\}$;
- 2) $J\text{-exp } M \leq n + \text{exp } C$.

Величины $\text{exp } C$ и $J\text{-exp } M$ можно оценить через характеристики генератора.

- 1) Из [2, с. 227] следует, что $\text{exp } C \leq m + j_1(m-2)$, тогда в соответствии с теоремой 3, п. 1

$$J\text{-exp } M \leq \max\{m, n + j_1\} + j_1(m-2).$$

- 2) Пусть среди чисел j_1, \dots, j_r, m имеется пара взаимно простых чисел, например $(j_1, j_2) = 1$, тогда из [3, теорема 1, б] следует, что $\text{exp } C \leq 2m + j_2j_1 - j_2 - 2j_1$. В этом случае в соответствии с теоремой 3, п. 1

$$J\text{-exp } M \leq \max\{n + j_1(m-1), 2m + j_2j_1 - j_2 - 2j_1\}.$$

Если, в частности, $j_1 > 2$ и $j_2 \leq \frac{m(j_1-2) + j_1}{j_1-1}$, то $\text{exp } C \leq j_1(m-1)$, то есть оценка теоремы 3, п. 2 точнее. Тогда в соответствии с теоремой 3, п. 2

$$J\text{-exp } M \leq n + 2m + j_2j_1 - j_2 - 2j_1.$$

ЛИТЕРАТУРА

1. Фомичев В. М. Методы дискретной математики в криптологии. М.: Диалог-МИФИ, 2010.
2. Сачков В. Н., Тараканов В. Е. Комбинаторика неотрицательных матриц. М.: ТВП, 2000.
3. Фомичев В. М. Оценки экспонентов примитивных графов // Прикладная дискретная математика. 2011. № 2(12). С. 101–112.

УДК 519.17

ДИСКРЕТНАЯ ДИНАМИЧЕСКАЯ СИСТЕМА НА ДВОЙНОМ ЦИРКУЛЯНТЕ С РАЗНЫМИ ФУНКЦИЯМИ В ВЕРШИНАХ

А. М. Нажмиденова

Исследована структура функционального графа дискретной динамической системы, состоящей из двух циркулянтов $G_{n,k}$ с различной ориентацией и мультипликативным отображением на одном циркулянте и аддитивным на другом. Описаны неподвижные точки, выведено рекуррентное соотношение для числа неподвижных точек и получена асимптотика этого числа, а также описаны висячие вершины и их число для частного случая $k = 2$.

Ключевые слова: *генная сеть, дискретная модель, регуляторный контур, циркулянт, функциональный граф, циклы, неподвижные точки, висячие вершины.*

Работа посвящена анализу функционирования дискретной модели генной сети. Характерной особенностью организации генных сетей является способность к саморегулированию через регуляторные контуры с положительными и отрицательными обратными связями. Процесс перераспределения концентраций веществ в регуляторном контуре может быть описан дискретной моделью, а строение регуляторных контуров может быть сформулировано в терминах ориентированных графов. В данной работе моделью является граф-носитель, состоящий из двух циркулянтов $G_{n,k}$ [1–3] противоположной ориентации, соответствующие вершины которых попарно сопряжены. Вершины графа-носителя имеют веса $x_0, x_1, \dots, x_{n-1}, y_0, y_1, \dots, y_{n-1}$ из конечного поля F_2 , где x_i соответствуют вершинам первого циркулянта, а y_i — вершинам второго. Набор $\tilde{w} = (x_0, \dots, x_{n-1}, y_0, \dots, y_{n-1}) \in F_2^{2n}$ назовем *состоянием системы*. В каждый момент времени состояние системы меняется и динамика его изменения определяется отображением

$$Func_{Mult,Add} : F_2^{2n} \rightarrow F_2^{2n},$$

где $Mult$ — мультипликативное отображение, действующее на вершинах первого циркулянта, и Add — аддитивное на вершинах второго, принимающие значения из F_2 в каждой вершине в зависимости от весов в тех k вершинах, дуги из которых входят в данную вершину.

Функциональным графом $G_{Mult,Add}$ называется орграф, вершинами которого являются наборы из F_2^{2n} , причём дуга из вершины \tilde{w} идёт в вершину \tilde{v} тогда и только тогда, когда $Func_{Mult,Add}(\tilde{w}) = \tilde{v}$.

Описаны неподвижные точки для произвольных n и k , а также выведено рекуррентное соотношение и асимптотика числа неподвижных точек.

Теорема 1. Число неподвижных точек S_n выражается рекуррентной формулой

$$S_n = S_{n-1} + S_{n-k}. \quad (1)$$

Для асимптотического поведения S_n справедливо

$$S_n \sim c_k R^n,$$

где c_k — константа, зависящая только от k , а $1 < R < 2$ — наибольший по модулю корень характеристического уравнения

$$\lambda^k - \lambda^{k-1} - 1 = 0$$

рекуррентного соотношения (1).

Для случая $k = 2$ доказана

Теорема 2. Число висячих вершин функционального графа равно $2^{2n} - 3^n$.

Получены необходимые и достаточные условия принадлежности набора циклу длины не более двух.

Теорема 3 (необходимое условие). В графе функционирования для цикла длины не более двух вида $(\alpha, \beta) \rightarrow (\gamma, \delta) \rightarrow (\alpha, \beta)$ выполнены условия $\gamma = \bar{\beta}$, $\delta = \bar{\alpha}$, где $\alpha, \beta, \gamma, \delta$ — наборы длины n .

Теорема 4 (достаточное условие). Если в наборе $\tilde{x} = (x_0, \dots, x_{n-1}, y_0, \dots, y_{n-1})$ для всех $i = 0, \dots, n - 1$ выполняются условия

- 1) если $x_i = 0$, то $y_{(i-1) \bmod n} = y_{(i+1) \bmod n} = 0$;
- 2) если $y_i = 1$, то $x_{(i-1) \bmod n} = x_{(i+1) \bmod n} = 1$,

и при этом $x_j = y_j$ для некоторого j , то \tilde{x} принадлежит циклу длины два.

ЛИТЕРАТУРА

1. Григоренко Е. Д., Евдокимов А. А., Лихошвай В. А., Лобарева И. А. Неподвижные точки и циклы автоматных отображений, моделирующих функционирование генных сетей // Вестник Томского государственного университета. Приложение. 2005. № 14. С. 206–212.
2. Evdokimov A. A. and Kutumova E. O. The discrete model of the gene networks regulatory loops with the threshold functions // Proc. 7th Int. Conf. on bioinformatics of genom regulation and structure. Novosibirsk, June 20–27, 2010. P. 155.
3. Харари Ф. Теория графов М.: Наука, 2003.

УДК 519.17

О Т-НЕПРИВОДИМЫХ РАСШИРЕНИЯХ СВЕРХСТРОЙНЫХ ДЕРЕВЬЕВ

Д. Ю. Осипов

Рассматривается один из способов построения оптимального расширения графа — Т-неприводимого расширения (ТНР). Приводится способ построения всех неизоморфных ТНР для подкласса сверхстройных деревьев — равнолучевых звезд.

Ключевые слова: граф, Т-неприводимое расширение, сверхстройные деревья, равнолучевые звезды.

Все понятия и определения взяты из работы [1].

Определение 1. Расширением n -вершинного графа G называется граф H с $(n + 1)$ вершинами, такой, что граф G вкладывается в каждый максимальный подграф графа H .

Простейшим примером расширения графа является его тривиальное расширение — соединение с одноэлементным графом (т. е. к графу G добавляется новая вершина, которая соединяется ребром с каждой вершиной графа G).

Возникает вопрос о получении такого расширения графа G , которое не содержит «лишних» ребер. Один из способов — конструкция минимального расширения графа [2], другой — его Т-неприводимого расширения [3].

Определение 2. Минимальным расширением графа G называется его расширение с минимальным количеством ребер.

В общем случае при построении минимального расширения возникает необходимость добавлять ребра в исходный граф, т. е. менять всю систему, моделируемую графом. Но иногда технически важно найти решение следующей задачи: построить оптимальное расширение данного графа, сохраняя его первоначальную конструкцию (т. е. не меняя связей внутри него). Существует следующая процедура:

- построить тривиальное расширение исходного графа;
- удалять из полученного графа рёбра до тех пор, пока выполняется свойство расширения.

Полученные графы назовем T -неприводимыми расширениями. Для произвольного графа количество неизоморфных ТНР неизвестно.

Покажем примеры ТНР для некоторых классов графов. Для n -вершинной цепи единственным ТНР является $(n + 1)$ -вершинный цикл. Для n -вершинного цикла единственным ТНР является тривиальное расширение исходного цикла.

На рис. 1 представлен граф, имеющий два неизоморфных ТНР.

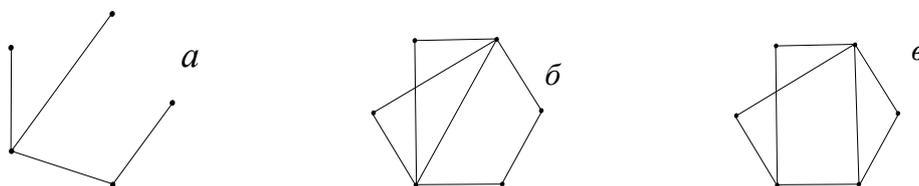


Рис. 1. Граф G (a) и два его неизоморфных ТНР (b и v)

Существует следующая нерешенная задача: для произвольного дерева построить все неизоморфные ТНР. Данная задача не решена и для произвольного сверхстройного дерева. Все неизоморфные ТНР для произвольной пальмы найдены в работе С. Г. Курносовой [4]. В настоящей работе найдены все неизоморфные ТНР для ещё одного подкласса сверхстройных деревьев — равнолучевых звезд.

Определение 3. Граф называется равнолучевой звездой с m лучами, каждый из которых состоит из n вершин, если $V = \{v_0, v_1^1, \dots, v_n^1, \dots, v_1^m, \dots, v_n^m\}$, $\alpha = \{v_i^j v_{i+1}^j : i = 1, \dots, n - 1; j = 1, \dots, m\} \cup \{v_0 v_1^j : j = 1, \dots, m\}$, где v_0 — центр равнолучевой звезды.

Теорема 1. Пусть граф S_n^m — равнолучевая звезда с m лучами, каждый из которых состоит из n вершин ($n \geq 2$). Тогда единственным ТНР для S_n^m является граф, полученный из тривиального расширения графа S_n^m удалением ребер wv_{n-1}^j , $j = 1, \dots, m$, где w — вершина, добавленная при построении тривиального расширения графа S_n^m .

ЛИТЕРАТУРА

1. Богомолов А. М., Салий В. Н. Алгебраические основы теории дискретных систем. М.: Наука, 2009.
2. Абросимов М. Б. Минимальные расширения объединения некоторых графов // Теоретические проблемы информатики и её приложений. 2001. № 4. С. 3–11.
3. Салий В. Н. Доказательства с нулевым разглашением в задачах о расширениях графов // Вестник Томского государственного университета. Приложение. 2003. № 6. С. 63–65.
4. Курносова С. Г. T -неприводимые расширения для некоторых классов графов // Теоретические проблемы информатики и её приложений. 2004. № 6. С. 113–125.

УДК 519.17

ОБ УПОРЯДОЧЕННОМ МНОЖЕСТВЕ СВЯЗНЫХ ЧАСТЕЙ МНОГОУГОЛЬНОГО ГРАФА

В. Н. Салий

Охарактеризованы многоугольные графы, для которых упорядоченное вложением множество абстрактных связных частей является решёткой.

Ключевые слова: *многоугольный граф, линейный граф, двоичный вектор, двойственность, упорядоченное множество, решётка.*

Под графом понимается пара $G = (V, \alpha)$, где V — конечное непустое множество и $\alpha \subseteq V \times V$ — отношение на нём. Элементы множества V называются вершинами графа, а пары, входящие в отношение смежности α , дугами.

Если $V' \subseteq V$ и $\alpha' \subseteq \alpha$, то граф $G' = (V', \alpha')$ называется частью графа G . В случае, когда $\alpha' = \alpha \cap (V' \times V')$, говорят, что G' является подграфом графа G .

Пусть $G = (V, \alpha)$ и $H = (U, \beta)$ — некоторые графы. Вложение графа G в граф H — это такое инъективное отображение $\varphi : V \rightarrow U$, что $(\forall v, v' \in V)((v, v') \in \alpha \implies (\varphi(v), \varphi(v')) \in \beta)$. Если $(\forall v, v' \in V)((v, v') \in \alpha \iff (\varphi(v), \varphi(v')) \in \beta)$, то говорят, что φ — сильное вложение G в H . Биективное сильное вложение (фактически наложение) $\varphi : V \rightarrow U$ по определению является изоморфизмом графа G на граф H . Если граф G вкладывается в граф H , то G изоморфен некоторой части графа H , а при сильном вложении — некоторому его подграфу.

Вершины v, v' графа G называются связанными, если $(\exists v_1, v_2, \dots, v_k \in V)((v, v_1) \in \alpha \cup \alpha^{-1} \& (v_1, v_2) \in \alpha \cup \alpha^{-1} \& \dots \& (v_k, v') \in \alpha \cup \alpha^{-1})$. Граф, в котором любые две вершины связаны, по определению является связным.

Маршрутом с началом v и концом v' называется последовательность примыкающих дуг $(v, v_1), (v_1, v_2), \dots, (v_k, v')$. Маршрут можно представить в виде перечисления проходимых вдоль него вершин: $vv_1v_2 \dots v_kv'$. Цепь — это маршрут, в котором все вершины разные. Цепь, состоящую из n дуг, обозначим через P_n и будем использовать её стандартную запись $P_n = v_0v_1 \dots v_n$. Если «склеить» концы цепи, получим n -звенный (n -вершинный) контур, который будем записывать в виде $C_n = v_1v_2 \dots v_{n-1}v_1$, считая v_1 выбранной начальной вершиной.

Под линейным графом длины n понимается всякий граф L , полученный переориентацией некоторых дуг цепи P_n . Многоугольным графом порядка n называется всякий граф M , полученный переориентацией некоторых дуг контура C_n .

Все связные части линейного графа являются его подграфами. В многоугольном графе M порядка n все связные собственные части с не более чем $n - 1$ вершинами являются линейными подграфами в M ; если же из M удалить какую-нибудь дугу, то получится линейный граф, являющийся частью, но не подграфом графа M .

Для многоугольного графа M через $\text{ASubc } M$ обозначим класс всех связных графов, допускающих вложение в M . Если $\mathcal{L} \in \text{ASubc } M$, то это означает, что все графы из \mathcal{L} изоморфны некоторой линейной части графа M или самому графу M . Класс $\text{ASubc } M$ упорядочивается отношением вложимости: если \mathcal{L}' и \mathcal{L}'' определяются соответственно линейными частями L' и L'' графа M , то $\mathcal{L}' \leq \mathcal{L}''$ по определению означает, что L' вкладывается в L'' .

В [1] автором охарактеризованы линейные графы L , для которых упорядоченное множество $\text{ASubc } L$ всех связных абстрактных подграфов является решёткой. Настоящее сообщение существенно опирается на идеи и методы, представленные в [1].

Пусть b — некоторый двоичный вектор. Двойственным для него называется вектор b^δ , получаемый из b так: компоненты вектора b надо записать в обратном порядке, а затем взаимно заменить в компонентах нули и единицы, т. е. осуществить преобразование $b \mapsto (b^{-1})'$. Например, для $b = 011100$ получим $b^\delta = 110001$. Понятно, что $b^{\delta\delta} = b$.

Под отрезками вектора понимаются блоки, состоящие из подряд идущих компонент этого вектора. Через $\text{ASubc } b$ обозначим совокупность всех попарно не двойственных отрезков двоичного вектора b . На множестве $\text{ASubc } b$ вводится порядок: $b' \leq b''$, если b' является отрезком в b'' или в b''^δ .

Двоичными векторами естественным образом кодируются линейные и многоугольные графы. Линейному графу L длины n соотносится двоичный n -мерный вектор $b = b(L)$ путём сопоставления каждой дуге графа символа 1, если при переориентации цепи $P = v_0 v_1 \dots v_n$ в граф L эта дуга оказалась направленной от v_0 к v_n , и символа 0 в противном случае. Например, для $L = v_0 \leftarrow v_1 \leftarrow v_2 \rightarrow v_3 \rightarrow v_4 \leftarrow v_5$ будет $b(L) = 00110$. С другой стороны, каждому n -мерному двоичному вектору b соответствует линейный граф $L = L(b)$ длины n , получающийся из цепи P_n переориентацией её дуг, согласованной в вышеуказанном смысле со значениями компонент вектора b . Так, для $b = 1011$ будет $L(b) = v_0 \rightarrow v_1 \leftarrow v_2 \rightarrow v_3 \rightarrow v_4$. Двоичным кодом для связного подграфа линейного графа L , очевидно, является отрезок вектора $b(L)$ или двойственного. Заметим, что двойственные векторы являются кодами изоморфных линейных графов. Будем считать, что $b(L)$ является лексикографически меньшим из них.

Пусть M — многоугольный граф, полученный из контура C_n переориентацией некоторых дуг. Выберем в C_n в качестве начальной вершину v_1 и построим n -мерный двоичный вектор b^1 , полагая $b_i^1 = 1$, если $(v_i, v_{i+1}) \in \alpha$ в M , и $b_i^1 = 0$, если $(v_{i+1}, v_i) \in \alpha$ в M (сложение в индексах — по модулю n). Аналогично построим вектор b^2 , считая начальной вершиной v_2 , и т. д. Выбрав из векторов b^1, b^2, \dots, b^n лексикографически минимальный, сопоставим его графу M и обозначим через $b(M)$. Например, для четырёхугольного графа $M = v_1 \rightarrow v_2 \leftarrow v_3 \leftarrow v_4 \rightarrow v_1$ получим $b^1 = 1001$, $b^2 = 0011$, $b^3 = 0110$, $b^4 = 1100$, и значит, $b(M) = 0011$. С другой стороны, каждому n -мерному вектору b соответствует n -угольный граф $M = M(b)$, получающийся из контура C_n переориентацией некоторых его дуг, согласованной в вышеуказанном смысле со значениями компонент вектора b . Например, для $b = 01001$ будет $M(b) = v_1 \leftarrow v_2 \rightarrow v_3 \leftarrow v_4 \leftarrow v_5 \rightarrow v_1$.

Лемма 1. Если M — многоугольный граф и b — соответствующий ему двоичный вектор, то упорядоченные множества $\text{ASubc } M$ и $\text{ASubc } b$ изоморфны.

Из леммы 1 следует, что упорядоченное множество $\text{ASubc } M$ абстрактных связных частей многоугольного графа M тогда и только тогда будет решёткой, когда решёткой является упорядоченное множество $\text{ASubc } b$ попарно не двойственных отрезков двоичного вектора b , кодирующего граф M .

При записи двоичных векторов в них группируются одинаковые компоненты и используется экспоненциальное обозначение: $01100110010 = 0(1^2 0^2)^2 10$ и т. п.

Теорема 1. Пусть M — многоугольный граф с n вершинами. Упорядоченное множество $\text{ASubc } M$ его абстрактных связных частей тогда и только тогда является решёткой, когда вектор $b = b(M)$ имеет один из следующих видов: 1) 0^n ; 2) $0^{n-1}1$, $n \leq 4$; 3) $0^{n-2}1^2$; 4) $(0^k 1^k)^l$ при $k \geq 1$, $l \geq 1$, $2kl = n$.

ЛИТЕРАТУРА

1. Саллий В. Н. Система абстрактных связанных подграфов линейного графа // Прикладная дискретная математика. 2012. № 2(16). С. 90–94.

УДК 519.7

ПРОСТОЕ ДОКАЗАТЕЛЬСТВО СИЛЬНОЙ РЕГУЛЯРНОСТИ ГРАФА КЭЛИ БЕНТ-ФУНКЦИИ¹

Н. Н. Токарева

Получено простое доказательство известного результата об описании класса бент-функций в терминах сильно регулярных графов.

Ключевые слова: бент-функции, сильно регулярные графы.

Бент-функции возникают в ряде криптографических приложений и широко исследуются. В частности, очень важной является задача описания бент-функций в алгебраических и комбинаторных терминах [1].

Пусть f — булева функция от n переменных. Через $\text{supp}(f)$ обозначим её носитель, т. е. множество всех двоичных векторов длины n , на которых функция f принимает значение 1. Рассмотрим граф Кэли $G_f = G(\mathbb{Z}_2^n, \text{supp}(f))$ булевой функции f . Вершинами графа являются все векторы длины n . Две вершины x, y соединяются ребром, если вектор $x \oplus y$ принадлежит множеству $\text{supp}(f)$. Граф G называется *сильно регулярным с параметрами* (v, k, λ, μ) , если он содержит v вершин, степень каждой вершины равна k и для любых двух вершин x, y число общих смежных им вершин равно λ или μ в зависимости от того, соединены вершины x, y ребром или нет. Булева функция f от чётного числа переменных n называется *бент-функцией*, если ее производная по любому ненулевому направлению y уравновешена, т. е. выполняется $\sum_{x \in \mathbb{Z}_2^n} (-1)^{f(x)+f(x+y)} = 0$.

В 2001 г. А. Бернаскони, Б. Коденотти и Дж. Ван-дер-Кам [2, 3] получили следующий результат, позволивший охарактеризовать множество бент-функций в терминах сильно регулярных графов.

Теорема 1. Булева функция f является бент-функцией тогда и только тогда, когда граф G_f является сильно регулярным, причем $\lambda = \mu$.

Доказательство было получено с помощью спектральной техники при исследовании графов, ассоциированных с булевыми функциями. Бент-функции при этом составили частный случай булевых функций с тремя различными спектральными коэффициентами. Недостатком такого доказательства служит его объёмность и малая наглядность: остаётся трудным объяснить по существу, почему же графы Кэли бент-функций (и только они) обладают таким примечательным свойством, как сильная регулярность. В связи с этим результат было трудно включить, например, в учебный курс.

В данной работе получено простое доказательство теоремы 1, непосредственно опирающееся на свойства бент-функций. Явно определены параметры сильно регулярных графов, соответствующих бент-функциям. Теорема 1 вытекает из следующих утверждений, доказательства которых теперь могут войти в любой учебный курс.

Утверждение 1. Граф Кэли G_f бент-функции f от n переменных сильно регулярный с параметрами $(2^n, 2^{n-1} \pm 2^{(n/2)-1}, \lambda = \mu = 2^{n-2} \pm 2^{(n/2)-1})$.

¹Работа поддержана грантами РФФИ № 11-01-00997, 12-01-31097.

Знаки « \pm » считаются согласованными, т. е. одинаковыми в обоих случаях.

Утверждение 2. Пусть G — сильно регулярный граф на 2^n вершинах, $n \geq 2$, такой, что $\lambda = \mu > 0$. Тогда он имеет параметры из утверждения 1.

ЛИТЕРАТУРА

1. Токарева Н. Н. Нелинейные булевы функции: бент-функции и их обобщения. Saarbrücken: LAP Lambert Academic Publishing, 2011.
2. Bernasconi A. and Codenotti B. Spectral analysis of Boolean functions as a graph eigenvalue problem // IEEE Trans. Computers. 1999. V. 48. No. 3. P. 345–351.
3. Bernasconi A., Codenotti B., and VanderKam J. M. A characterization of bent functions in terms of strongly regular graphs // IEEE Trans. Computers. 2001. V. 50. No. 9. P. 984–985.

УДК 519.248, 519.176

АСИМПТОТИКИ ВЕРОЯТНОСТЕЙ СВЯЗНОСТИ ПАР ВЕРШИН ГРАФА¹

Г. Ш. Цициашвили, М. А. Осипова, А. С. Лосев

Для графов с низконадёжными ребрами построена асимптотика вероятности связности любой пары его вершин. Параметрами полученного соотношения являются характеристики кратчайших путей графа, для вычисления которых разработаны модификации классических алгоритмов. Проведенный вычислительный эксперимент продемонстрировал преимущества предложенных алгоритмов.

Ключевые слова: кратчайший путь, вероятность связности, вычислительная сложность.

Для случайных графов с низконадёжными рёбрами построен удобный в реализации алгоритм вычисления вероятности связности любой пары его вершин на основе доказанного асимптотического соотношения. Для параметров полученного соотношения (характеристик кратчайших путей) разработаны модификации классических алгоритмов. Особенностью предлагаемых алгоритмов является тот факт, что в них не требуется перечислять кратчайшие пути между узлами, нужно лишь определить их количество. Ещё одним существенным фактором упрощения вычислений является рассмотрение графов с ограниченным диаметром, которые в последние годы вызывают большой теоретический и практический интерес. Проведенный вычислительный эксперимент подтвердил быстроедействие построенной процедуры определения вероятности связности по сравнению с методом Монте-Карло.

Рассмотрим неориентированный связный простой граф G с множеством узлов U и множеством рёбер V . Предположим, что каждое ребро v графа G с вероятностью $p(v)$ работоспособно, причём все рёбра функционируют независимо. Обозначим $D(i, j)$ минимальное число рёбер в путях, соединяющих узлы i, j графа G , $C(i, j)$ — число путей с $D(i, j)$ рёбрами. Для вероятности связности $P_{ij}(G)$ узлов i, j графа G доказаны следующие утверждения.

Теорема 1. Если $p(v) = h$, $v \in V$, то

$$P_{ij}(G) \sim C(i, j)h^{D(i, j)}, \quad h \rightarrow 0. \quad (1)$$

¹Работа поддержана грантом РФФИ № 12-01-00114-а.

Следствие 1. Если $p(v) = h$, $v \in V$, то

$$\min_{1 \leq i, j \leq n} P_{ij}(G) \sim Ch^D, \quad h \rightarrow 0,$$

$$D = \max_{1 \leq i, j \leq n} D(i, j), \quad C = \min_{(i, j): D(i, j)=D} C(i, j).$$

Для нахождения всех элементов матриц $\|D(i, j)\|_{i, j=1}^n$, $\|C(i, j)\|_{i, j=1}^n$ асимптотической формулы (1) построены модификации известных в теории графов алгоритмов (в том числе Флойда — Стейнберга). Такая процедура является более экономичной, чем последовательное определение элементов этих матриц, имеет вычислительную сложность $O(n^3 \ln n)$ для матрицы $\|D(i, j)\|_{i, j=1}^n$ и $O(n^4)$ для матрицы $\|C(i, j)\|_{i, j=1}^n$. Зная матрицу $\|D(i, j)\|_{i, j=1}^n$, можно вычислить диаметр D графа G . Для сетей с ограниченным диаметром D сложность вычисления матриц $\|D(i, j)\|_{i, j=1}^n$, $\|C_s(i, j)\|_{i, j=1}^n$ составляет $O(n^3)$ арифметических операций.

На основе построенного алгоритма для вероятности связности любой пары вершин графа был проведён вычислительный эксперимент. Зададим граф G матрицей смежности

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Полагаем, что работоспособность его рёбер равна $p(v) = h = 0,01$. Используя модифицированные алгоритмы, вычислим

$$\|D(i, j)\|_{i, j=1}^6 = \begin{pmatrix} 0 & 1 & 2 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 & 2 & 1 \\ 2 & 1 & 0 & 1 & 2 & 2 \\ 1 & 2 & 1 & 0 & 1 & 2 \\ 2 & 2 & 2 & 1 & 0 & 1 \\ 1 & 1 & 2 & 2 & 1 & 0 \end{pmatrix}, \quad \|C(i, j)\|_{i, j=1}^6 = \begin{pmatrix} 0 & 1 & 2 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 & 1 & 1 \\ 2 & 1 & 0 & 1 & 1 & 1 \\ 1 & 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 2 & 1 & 0 \end{pmatrix}.$$

Результаты вычислений вероятностей связности пар вершин $P_{ij}(G)$, $1 \leq i, j \leq 6$, по формуле (1) и методом Монте-Карло (обозначим их $P_{ij}^*(G)$) при 10^6 итераций представлены ниже:

$$\|P_{ij}(G)\|_{i, j=1}^6 = \begin{pmatrix} 1 & 0,01 & 0,0002 & 0,01 & 0,0002 & 0,01 \\ 0,01 & 1 & 0,01 & 0,0002 & 0,0001 & 0,01 \\ 0,0002 & 0,01 & 1 & 0,01 & 0,0001 & 0,0001 \\ 0,01 & 0,0002 & 0,01 & 1 & 0,01 & 0,0002 \\ 0,0002 & 0,0001 & 0,0001 & 0,01 & 1 & 0,01 \\ 0,01 & 0,01 & 0,0001 & 0,0002 & 0,01 & 1 \end{pmatrix},$$

$$\|P_{ij}^*(G)\|_{i, j=1}^6 = \begin{pmatrix} 1 & 0,01035 & 0,000203 & 0,010027 & 0,000192 & 0,010205 \\ 0,01035 & 1 & 0,010001 & 0,000198 & 0,000095 & 0,009764 \\ 0,000203 & 0,010001 & 1 & 0,010083 & 0,000094 & 0,000103 \\ 0,010027 & 0,000198 & 0,010083 & 1 & 0,010051 & 0,000208 \\ 0,000192 & 0,000095 & 0,000094 & 0,010051 & 1 & 0,009973 \\ 0,010205 & 0,009764 & 0,000103 & 0,000208 & 0,009973 & 1 \end{pmatrix}.$$

Время счета по формуле (1) составило 10 с, методом Монте-Карло — 6 ч.

Более подробное изложение представленных результатов можно найти в [1].

ЛИТЕРАТУРА

1. Цициашвили Г. Ш., Осипова М. А., Лосев А. С. Асимптотика вероятности связности графа с низконадёжными рёбрами // Прикладная дискретная математика. 2013. № 1(19). С. 93–98.

Секция 5

**МАТЕМАТИЧЕСКИЕ ОСНОВЫ
ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

УДК 004.43, 004.056

**CRYPTOGRAPHIC EXTENSION
OF RUSSIAN PROGRAMMING LANGUAGE**

G. P. Agibalov, V. B. Lipsky, I. A. Pankratova

An extension of the Russian programming language LYaPAS called LYaPAS-T is presented. The extension concerns the size of operands and the set of elementary operations over them. It is caused by the need of trustworthy and effective soft and hard implementations of contemporary cryptographic algorithms in secure computer systems applied for the logical control of critically important objects such as cosmic systems, nuclear weapons, energetic plants, submarines, etc. A LYaPAS-T compiler generating a load module for operating system Linux is presented too.

Keywords: *Russian programming language, cryptographic extension, LYaPAS-T, compiler.*

Introduction

Here by the Russian programming language is meant the algorithmic language LYaPAS elaborated at the beginning of the 1960th years at Tomsk State University (Russia) by the leadership of A. D. Zakrevskij and designed for the representation of logical combinatorial algorithms solving the problems of applied discrete mathematics appearing in the synthesis of discrete automata [1, 2]. The name of Russian programming language was given to it by American scientists [3]. Up to 1990th years, LYaPAS was applied in USSR [2], USA [4], Germany, Poland, Czechoslovakia and other countries. This time, LYaPAS is successfully reanimated at Tomsk State University by the Information Security and Cryptography Department especially for elaborating the trustworthy system and applied software destined for the computer-aided design of secure logical control computer systems and for the secure and effective implementation of cryptographic algorithms [5]. Among many programming languages known today, LYaPAS seems to be the most appropriate one for these purposes.

At the same time, there is an essential and perhaps single drawback of LYaPAS — the lack of many elementary operations that are widely used in contemporary cryptographic algorithms: for the long integer arithmetic, for calculating in many-dimensional spaces over finite fields and rings, for solving combinatorial problems on large sets, etc. By the way, this drawback is usual for all other programming languages including ones being much younger than LYaPAS. In some of them, the drawback is got over by writing classes of long integers, large discrete functions and others. As for LYaPAS, this its lack is more effectively got over by extending the language itself by spreading elementary operations in LYaPAS for logical complexes and by adding to it some new elementary operations defined for variables and logical complexes with the wanted purposes. The last version of LYaPAS known as LYaPAS-M [6, 7] slightly revised and then exTended in such a way is called LYaPAS-T.

The revision of LYaPAS-M concerns the symbol alphabet of the language and the arithmetic operations of multiplication and division for integers. The result of the revision is called vLYaPAS (from reVised LYaPAS). In it, small Latin letters are used instead of capital Russian ones, symbols of some operations are replaced by other more proper ones, and multiplication and division operations are defined saving the values of overflow and remainder respectively. For keeping them, a special variable — Z is inserted in the language.

The objective of the paper is to present an information about vLYaPAS, its extension LYaPAS-T motivated by the requirements of cryptographic algorithms, and a LYaPAS-T compiler generating code in the executive file format of the operating system (OS) Linux.

The project of a processor implementing LYaPAS-T programs in hardware, and a pre-processor translating LYaPAS-T programmes to executive code for the processor are presented in [8], the program in vLYaPAS-T representing the cryptographic algorithm AES is demonstrated in [9].

1. Revised LYaPAS

A program in vLYaPAS is a series of sentences each starting with a pair $\$s$ where s is a non-negative integer called the number of the sentence. Every sentence is a sequence of operations applied to operands.

1.1. Operands

Operands in vLYaPAS are constants, variables, complexes and complex elements. They are used for representing non-negative integers, Boolean vectors, Unicode symbols and sequences of them. In vLYaPAS, non-negative integers are bounded by $2^{32} - 1$, the length of Boolean vectors does not exceed 32. Boolean vectors of length 32 are also called words. Components in a Boolean vector are numbered beginning with 0 in the direction from the right to the left end, and the vector itself can be considered as a non-negative integer (written in the binary form). A Boolean vector of any length $n \geq 1$ with only one component 1 is called a unit or identical vector.

There are natural, unit and symbol constants. Natural constants are written as decimal, hexadecimal, octal or binary numbers. A unit constant is a Boolean vector with only one component 1. It is denoted by I_i if the number of component 1 in it equals i . Symbol constant is a sequence of Unicode symbols.

Variables in vLYaPAS take values of Boolean vectors of length 32. The number of variables equals 27. They are denoted by letters a, b, \dots, z, Z where Z is used for specialized purposes mentioned above. Besides, there is a virtual variable also called the own or internal variable in LYaPAS. Unlike the other variables, it is not written in LYaPAS programs, but it appears in them in implicit way as a result of any elementary operation and can be used as operand by any next operation in the program. For convenience of exposition, this variable is accepted to name τ .

Complex is a linearly ordered set of elements being symbols in a symbol complex or Boolean vectors of length 32 in a logical complex. Every complex has its own unique number from the series 0, 1, 2, ... The logical or symbol complex having number i is denoted by the symbol L_i or F_i respectively. The real number and the greatest number of elements in a complex are the parameters of the complex and are called its cardinality and capacity respectively.

Unlike the other programming languages, the value types in vLYaPAS are not fixed. The value type (an integer or a vector) for a constant, variable and complex element is determined by the type of an operation (arithmetic or logical respectively) which is applied to this operand.

1.2. Operations

In vLYaPAS, there are elementary, complex, input-output and macro operations. Elementary operations are, in turn, of the following classes:

- value transfer: O — assigning 0s, $\bar{}$ — assigning 1s, x — assigning x , \Rightarrow — assigning τ , X — assigning an output value from a computer pseudorandom number generator (PRNG), \Rightarrow X — assigning τ to the seed of PRNG, T — assigning computer time, $=(x, y)$ — value swop;
- logical: \neg — negation, \vee — disjunction, $\&$ — conjunction, \oplus — exclusive disjunction, $<$ — left shift, $>$ — right shift;
- arithmetic: ! — right 1 number calculation, % — Boolean vector weight calculation, + — modulo 2^{32} addition, — — modulo 2^{32} subtraction, * — modulo 2^{32} multiplication ($Z :=$ overflow), : — double number division ($Z :=$ remainder), / — integer division quotient ($Z :=$ remainder), ; — integer division remainder ($Z :=$ quotient), Δ — increase by 1, ∇ — decrease by 1;
- transition: \rightarrow — unconditional transition, \leftrightarrow — transition under condition $\tau = 0$, \mapsto — transition under condition $\tau \neq 0$, $\rightarrow(x \diamond y)$ — transition under relation $\diamond \in \{=, \neq, <, >, \leq, \geq\}$, $\rightarrow:$ — transition with return, $\rightarrow!$ — return back, $\rightarrow(x)$ — transition in case if the running time exceeds x ; $\rightarrow Xabc$ — ones enumeration: if $a = 0$ then $b := 0$ and $\rightarrow c$, otherwise right 1 in a is replaced by 0, its number $\Rightarrow b$ and next operation;
- {assembly program} — assembly insertion (the assembly program pointed between { and } is executed).

Among complex operations there are forming (creating) a complex of a given cardinality, annihilation, decreasing complex capacity up to cardinality, complex clearance (without changing cardinality), addition of symbols to symbol complex, element insertion, element exclusion, subcomplex copying to another complex.

In future a complex created with a constant or variable capacity will be called respectively static or dynamic one.

Note that the operation of taking subcomplex existing in LYaPAS is not included in vLYaPAS because of its potential insecurity.

Input-output operations are the following: $/F>C$ — output of symbol complex F to console; $/'\zeta'>C$ — output of symbol constant ζ to console; $/F<C$ — symbol constant input from keyboard: symbols are added to a symbol complex F, complex cardinality is increased.

Macro operations are calls for subprograms with the given external (input/output) parameters.

2. LYaPAS extension

2.1. Long arithmetics

Natural constants in LYaPAS-T are integers $0, 1, \dots, 2^n - 1$ where n is a multiple of 32 and depends on actual implementation of LYaPAS-T. Nowadays the value $n = 2^{14}$ seems to be quite sufficient for contemporary cryptographic applications.

By letting δ be 2^{32} a natural constant c may be expressed by the following series:

$$c = c_0 + c_1\delta + c_2\delta^2 + \dots + c_{r-1}\delta^{r-1} \quad (1)$$

for some $r > 0$ and $c_i \in \Omega = \{0, 1, \dots, 2^{32} - 1\}$, $i = 0, 1, \dots, r - 1$. In their standard binary representation, the elements of the set Ω are Boolean vectors of length 32. Therefore in

LYaPAS-T, the sequence c_0, c_1, \dots, c_{r-1} is represented by a logical complex L_j of length r with c_i being the i th element of L_j .

All operations defined in vLYaPAS for variables can be used in LYaPAS-T for logical complexes. In case of arithmetic operation the sequence of complex elements is considered as a natural constant c expressed by the series in (1). Different operands for an arithmetic operation may be of different lengths and types (one of them — a variable, another — a complex). In case of logical operation the complex value is considered as a Boolean vector being the concatenation of the complex elements. Logical complexes of cardinality $n/32$ with values being unit vectors are unit constants $I_i, i = 0, 1, \dots, n - 1$, in LYaPAS-T.

2.2. Plurality of own variable

So, unlike LYaPAS, there are two types of operands for elementary operations in LYaPAS-T: variables of the length of one word and logical complexes of different lengths — from 1 to $n/32$ words. Accordingly, in LYaPAS-T, there are two types of the own variable — prime and complex. The first one is the traditional τ from LYaPAS. It has the length of one word and may take the values of any variable of the language. In any implementation of LYaPAS-T, soft or hard, it is kept in a processor register. The own variables of the 2nd type take values of logical complexes and have their lengths. In hard implementation of LYaPAS-T, each of them is kept in one and the same register of the maximal possible length — n . In soft implementation of LYaPAS-T, to exclude time-spending operations for transferring complex between a register and data memory, it is expediently, for the time of executing a series of operations beginning with the address to a logical complex, to pass the role of the complex own variable directly to this complex and to keep it in data memory where the complex is kept.

2.3. Additional operations

In addition to operations in vLYaPAS, the extension LYaPAS-T contains some new logical operations used in cryptographic algorithms including the following ones:

- 1) $\updownarrow L$ — permutation: components of Boolean vector τ are re-arranged according to the order of their serial numbers pointed in a logical complex L ;
- 2) $-(a, b)$ — projection: the part of Boolean vector τ with components having numbers in an interval (a, b) is chosen;
- 3) $\uparrow b(i)$ — insertion: a Boolean vector b is put in τ before the i th element;
- 4) $\downarrow(a, b)$ — reduction: the part of Boolean vector τ with components having numbers in an interval (a, b) is deleted from the vector;
- 5) $| b$ — concatenation: a Boolean vector b is added to τ ;
- 6) $\ll i$ or $\gg i$ — left or right cyclic shift: Boolean vector τ is cyclically shifted i bits left or right respectively;
- 7) $\geq \varkappa(a)$ — most element: a maximal element is found in a complex \varkappa whose elements are considered as non-negative integers, its value is given to τ , its number — to a ;
- 8) $\leq \varkappa(a)$ — least element: analogously.

As for arithmetic operations modulo N (for any natural N) such as addition (mod N), multiplication (mod N), exponentiation (mod N) and others which are widely used in cryptography, there is no real possibility to include them in the list of the elementary operations in LYaPAS-T because of the existence of many algorithms implementing them with the different efficiencies in many cases. Instead, it is decided to implement these algorithms in an assembly language and (or) in LYaPAS-T and to include them in the LYaPAS-T library for applying them by users in LYaPAS-T programs as subprograms.

3. LYaPAS-T compiler

3.1. What is this?

For short, any LYaPAS-T program and its subprograms are called L-program and L-subprograms respectively. For being implemented by a computer, an L-program should be used as a parameter of LYaPAS-T compiler that converts it to a load module for the OS Linux.

The compiler starts under the command line of OS Linux

```
>$ lc <prog>.l
```

where `<prog>.l` is the name of a file with the L-program being a list of L-subprograms. (It is recommended to call an L-program file using the extension `l` but it is not necessarily.) The first L-subprogram in the list is the head one. OS Linux transfers the control to it after the file loading to RAM. The order of the other L-subprograms in the list does not matter. The file may contain not all necessary L-subprograms. In this case the compiler looks for them in the file `libl0.l` being the user library. It is the library where it is convenient to keep the often used L-subprograms.

The result of the compiler working is a load module which is kept with the name `<prog>` (without extension) at the same folder where the L-program is located. For executing it the following command is used:

```
>$ ./<prog>
```

Compiler is written in C++ with the using the library of regular expressions making it absolutely simple and transparent.

3.2. Load module structure

The load module consists of two segments — program segment (`.text`) and data segment (`.data`). In turn, the first one consists of subprograms generated by the compiler for L-subprograms called in the process of the L-program execution. The data segment contains: the current address in the memory for placing new complexes, and the bound of memory granted for complexes by OS; the current state of the PRNG; unit constants; the weights of all Boolean vectors in $\{0, 1\}^8$ (need for the implementation of the operation `%`); and all symbol constants met in the L-program.

3.3. Memory organization

For every L-subprogram, all its local variables, beginnings, cardinalities and capacities of its local complexes are placed in a stack forming a frame of 1420 bytes. An access to the local data is made by using fixed shifts from the frame beginning (register `ebp`). For the frame of a parental L-subprogram, the value of the register `ebp` is also saved in the frame. So, the list of frames is created for the L-subprograms called.

The local complexes of an L-subprogram are placed in a heap. The address of the free section in the heap at the moment of calling for the L-subprogram is also kept in the frame.

The operation of creating local complex is accompanied with the control of the free section of a necessary size. It is done by comparing the values of the complex capacity, address of the free area, and the memory bound granted by OS for complexes. In the case of enough place, the address of the complex beginning takes the value of the free section address, and the free section address is increased by the value of the complex capacity. If the place is not enough then an appeal to OS is made for increasing the accessible memory bound.

Under this organization, the memory is protected against attacks through the stack or heap overflow because, first, buffers (complexes) are taken away from the stack, and there is

no possibility to rewrite the return address, and, second, there are no operations for setting memory free by means of OS.

BIBLIOGRAPHY

1. LYaPAS, a Programming Language for Logic and Coding Algorithms / eds. M. A. Gavrilov and A. D. Zakrevskii. New York, London: Academic Press, 1969. 475 p.
2. *Toropov N. R.* Programming language LYaPAS // Applied Discrete Mathematics. 2009. No. 2(4). P. 9–25. (in Russian).
3. *Nadler N.* User group for Russian programming language // IEEE, Newsletter for Computer-Aided Design. 1971. Iss. 3.
4. *Charles J. and Albright Jr.* An Interpreter for the Language LYaPAS. University of North Carolina at Chapel Hill: Department of Computer Science, 1974. 125 p.
5. *Agibalov G. P.* To reanimation of Russian programming language // Applied Discrete Mathematics. 2012. No. 3(17). P. 77–84. (in Russian).
6. *Zakrevskij A. D. and Toropov N. R.* Programming system LYaPAS-M. Minsk: Nauka i Technika, 1978. 240 p. (in Russian).
7. *Toropov N. R.* Dialogue programming system LES. Minsk: Nauka i Technika, 1985. 263 p. (in Russian).
8. *Agibalov G. P., Lipsky V. B., and Pankratova I. A.* Project of hardware implementation of Russian programming language // Applied Discrete Mathematics. Application. 2013. No. 6. P. 98–102.
9. *Broslovskiy O. V.* AES in LYaPAS // Applied Discrete Mathematics. Application. 2013. No. 6. P. 102–104.

УДК 004.43, 004.056

PROJECT OF HARDWARE IMPLEMENTATION OF RUSSIAN PROGRAMMING LANGUAGE

G. P. Agibalov, V. B. Lipsky, I. A. Pankratova

The projects of a LYaPAS-T processor implementing the programming language LYaPAS-T in hardware and of a preprocessor translating LYaPAS-T programs into the executive code of the processor are presented. It is also told that for a LYaPAS-T subset containing neither subprograms, nor operations over complexes and long operands, the architecture of the processor was described in VHDL, tested by means of a computer simulation, and implemented in a programmable logical integrated circuit obtained with the help of a computer-aided design.

Keywords: *Russian programming language, LYaPAS-T, hardware implementation, LYaPAS-T preprocessor.*

In [1] a cryptographic extension LYaPAS-T of Russian programming language and a compiler for its implementation in software were presented. The objective of this paper is to present an information about the project of a processor implementing LYaPAS-T programs in hardware, and a preprocessor translating LYaPAS-T programs to executive code for the processor.

1. Parameters

In LYaPAS-T implemented in hardware, the operand length, the largest number of a complex and the maximal quantity of subprograms in the hierarchical structure of a program are assumed to be bounded by natural n , m and k respectively. Hence, the quantities of

different complexes (logical and symbol) and variables in a LYaPAS-T program executed by a processor do not exceed, respectively, mk and $27k$. Nowadays the values like $m = 64$, $k = 128$ are quite sufficient for the majority of practical algorithms.

2. Executive code

For being executed by LYaPAS-T processor, a LYaPAS-T program should be preliminarily represented by a sequence of instructions in the executive code (called LE-code) for the processor. Each instruction in it has fields containing information about operation code, operand type (constant or not, complex type — logical, symbol, static or dynamic if the operand is a complex or its element) and complex and variable addresses in the data memory. The field for complex address is used if the operation deals with the data of the form $\varkappa v$ or L. In the first case the address of static complex \varkappa being, by definition, the address of the first element in \varkappa is explicitly written in this field, and the address of the variable v is written in the field for variable address. In the second case the field of the complex address contains the address of the complex L, and the field of the variable address remains empty (equals 0). In all other cases the field of the complex address is empty. The same is true for dynamic complex with the following exception: the field of its address contains not the complex address itself but the address where it is kept.

3. Architecture

The architecture of the processor implementing LYaPAS-T in hardware consists of three units: Memory, Arithmetic Logical Unit (ALU) and Control Device (CD).

3.1. Memory

The Memory is divided into two segments — IM (Instruction Memory) and DM (Data Memory) used to store, respectively, a sequence of instructions in LE-code representing a LYaPAS-T program P and data for it — unit constants I_i , complexes and variables for every subprogram in the hierarchical structure of the program P . Accordingly, for P with k subprograms, DM is conditionally divided into four sections: section I — for keeping vectors $I_i, i = 0, 1, \dots, n - 1$; sections C and G divided into k subsections C_j and G_j — for keeping respectively static and dynamic complexes in j th subprogram; and section W also divided into k subsections W_j — for keeping local variables a, b, \dots, z, Z belonging to j th subprogram, and parameters and addresses of all complexes in j th subprogram.

Sections I, C, W form so called static memory and section G — the dynamic memory of the processor. The data allocation in the first is made by the L-preprocessor (before the execution of the program P), in the second — by the processor itself (during the execution of P).

Instructions in IM and data in DM are disposed compactly, with no gaps and in the order of section enumeration: I, C, W, G . The quantity of engaged elements in the subsection G_j is fixed as a value of an element in W_j . The address of this element is denoted a_j . Its value is the least address of free element in G_j . Particularly, before the beginning of the processor work addresses a_j for all j equal the number of elements in the static memory.

Static complexes being created in j th subprogram are placed in C_j by L-preprocessor pointing for them parameters and addresses in W_j . The cardinality and the address in W_j of a dynamic complex are remained by L-preprocessor undetermined. A dynamic complex being created in j th subprogram with the cardinality equaled the value of a variable ξ is placed in G_j as an array of a certain size with the address of the initial (first) element written in the element of W_j whose address is a_j . It is done in the same way by the processor

at the moment when it executes the given operation: parameters and the address of the complex are stored in W_j , the value of the element having address a_j is increased by ξ .

3.2. ALU

The unit ALU consists of three registers τ , Z and O of the maximal possible length n called the Registers of Common Use (CURs) and destined for keeping, respectively, variables τ , Z and operand being read from DM, and also of Operational Devices (OD) and a circuit CEA (Complex Element Address) for determining the address in DM of a complex element.

Operational devices are used for executing arithmetic and logical operations in LYaPAS-T over operands and with the operation results represented in registers τ , Z and O .

For a complex element $\varkappa v$, its address in DM is computed by the circuit CEA as $\theta = \varphi + 4v$ for logical \varkappa or as $\theta = \varphi + v$ for symbol \varkappa where φ is the address in DM of the first element in \varkappa (also called the address of the complex \varkappa itself).

3.3. Control device

The main functions of CD are the following: to receive and analyse information signals of other units, to select the next instruction from IM, to identify the operation code in it, to determine the operand address in DM and the next instruction address in IM, and to form and send control signals to other executive units. For carrying out these functions, CD contains Instruction Counter (IC), Instruction Register (IR) and two decoders — Address Decoder (ADec) and Operation Decoder (ODec).

4. Functioning algorithm

1. CD selects from IM an instruction at the address pointed in IC and writes it to IR.
2. ODec decodes the contents of the operation code field, ADec decodes the contents of type and operand (complex and/or variable) address fields in IR.
3. CD takes the information from ODec and ADec, generates the signals either for selecting from DM (possibly by means of the circuit CEA) an operand (constant, variable, logical complex or complex element) at the corresponding address and writing it into one of the CURs, or, if the operand is a constant given explicitly in the instruction, for writing it to the register O or to IC.
4. If the operation in the instruction is related to the functional type being a logical or arithmetic one, CD generates a signal initiating the corresponding OD.
5. Initiated OD fulfils the operation of the instruction in IR, and CD writes results into CURs according to the operation.
6. If the operation code is of the transition type, the value of the variable address field in the instruction in IR is written to IC; otherwise the state of IC is increased for selecting the next instruction from IM.
7. If the instruction in IR implies creation of a dynamic complex with a variable capacity ξ in a j th subprogram, the cardinality 0 and capacity ξ of this complex are written at addresses of its parameters in W_j , and the value at address a_j is stored to W_j as the address of this complex in G_j and then it is increased by ξ .

In this algorithm, the work of the control device CD is formally described by the finite automata model.

5. L-preprocessor

For translating LYaPAS-T program into LE-code, a special compiler (called L-preprocessor) is used. It can be written in any programming language (for example in LYaPAS-T)

and run on any computer provided with a proper compiler. The following is the sequence of main actions which L-preprocessor should make over a LYaPAS program P :

1) for every subprogram in the hierarchical structure of the program P , to prescribe to it a unique number from the series $1, 2, \dots, k$;

2) for every $j = 1, 2, \dots, k$, to give a unique address in W_j to every local variable and to every static local complex met in the j th subprogram of P ;

3) for all $i, j \in \{1, 2, \dots, k\}$ where $i \neq j$, for every call for j th subprogram from i th subprogram in P , the real parameters, pointed in the call, to substitute for the corresponding abstract (external) parameters in the text of j th subprogram; for every dynamic complex in j th subprogram, the address in W_j , where the address of this complex in G_j is written, to substitute for the name of this complex in the text of j th subprogram; and the sequence of instructions $a_i, \Rightarrow a_j$ and the text of j th subprogram to substitute for the call itself in P ;

4) every other operation in P to replace with the equivalent sequence of instructions in LE-code — either directly or through the intermediate replacement by the series of unary operations each having not more than one operand being different from τ .

6. Alternative variant

In the alternative variant of the processor, the segment IM saves the executive codes both of a head program and of all subprograms in its hierarchical structure. In this case, if the program P contains the call for a subprogram S , the L-preprocessor substitutes for this call in P an instruction A with the operation code of the transition to the address of the subprogram S executive code in IM, and, at the end of this code, it writes an instruction with the operation code of return back, that is, of transition to the address of an instruction in IM next to A . Besides, when executing the program P , the processor, before the execution of the instruction A , transfers the values of the input operands of the subprogram S , pointed in its call, at the corresponding addresses in DM, written in its executive code, and, before returning back to the instruction next to A , transfers the results of the subprogram code work at the addresses of the corresponding output operands, pointed in its call.

The transferring the values of the program operands decreases the rate of the processor work in comparison with its basic variant, but essentially reduces the necessary size of the segment IM. At the same time, the absence of transferring between operands in the basic variant does not guarantee the value integrity for the input operands of a subprogram if it is not provided by the programmer.

7. Applications

There are, at least, two possible applications of LYaPAS-T processor: as a cryptoprocessor and as a control processor. In the first case the segment IM is filled in LE-code of a program in LYaPAS-T representing a cryptographic algorithm, and data for it (a plaintext, a ciphertext, a key and others) are written in the segment DM. In the second case IM and DM are used for storing, respectively, LE-code of a LYaPAS-T program destined for the secure control of a critically important object (cosmic, energetic, transport, etc.) and data for this program.

8. vLYaPAS subset processor

Let L_1 be the vLYaPAS subset that includes all operations at the first level of vLYaPAS and does not contain (calls for) subprograms and operations over complexes. For the first time, a processor architecture for L_1 (also called L_1 -processor) was elaborated and

implemented in VHDL in 2012 by S. E. Soldatov, a student of the Information Security and Cryptography Department of Tomsk State University. For preliminary verification, all individual units in L_1 -processor and its architecture on the whole were simulated by means of the program product ModelSim PE Student Edition 10.1d. Besides, the programmable logical integrated circuit of L_1 -processor was synthesized with the help of the computer-aided design system ISE WebPACK 9.2i by Xilinx. The maximal operating frequency of the circuit equals 50 MHz which is equivalent to the circuit delay of 20 ns. The size of the circuit is the third of the size of Nexys2 FPGA debugging board by Digilent Inc.

This result shows that the implementation in hardware of the processor for LYaPAS-T is the quite real affair promising trustworthy means for effective performance of cryptographic and other combinatorial algorithms.

BIBLIOGRAPHY

1. Agibalov G. P., Lipsky V. B., and Pankratova I. A. Cryptographic extension of Russian programming language // Applied Discrete Mathematics. Application. 2013. No. 6. P. 93–98.

УДК 004.43, 004.056

AES IN LYAPAS

O. V. Broslavskiy

Programs in vLYaPAS representing the encryption and key expansion algorithms for symmetric block cipher AES are presented.

Keywords: *AES, LYaPAS.*

The objective of the paper is to present the description of the AES encryption and key expansion algorithms [1, 2] in the revised Russian programming language vLYaPAS [3]. The presented programs show the compactness, transparency and effectiveness of cryptographic algorithm representations in the language which was originally aimed at the representation of logical synthesis algorithms. It is assumed that the number of the cipher rounds is 10, and the lengths of the cipher block and key equal 128 bits. A ciphertext block is considered as a 2-measured array of 4×4 bytes. It is called a *state* and is represented by a logical complex of cardinality 4 whose elements are the rows of the state.

Further, the texts of the head programs and their subprograms are given. The external parameters in them are the following: L1 — the state (with the initial value equaled a plaintext block); L2 — the array of eleven 128-bit round keys (the complex of cardinality 44); L3 — ciphertext block; L4 — substitution table (S-box) for the operation of byte substitution; L5 — private key.

Encryption of a block

Encrypt(L1, L2, L4/L3)

*AddRoundKey(L1, L2, 0/L3) 0i

§1 $\Delta i \oplus 10 \leftrightarrow 2$

*SubBytes(L3, L4/L3)

*ShiftRows(L3/L3)

*MixColumns(L3/L3)

*AddRoundKey(L3, L2, i/L3) $\rightarrow 1$

§2 *SubBytes(L3, L4/L3)

*ShiftRows(L3/L3)

*AddRoundKey(L3, L2, 10/L3) **

Addition modulo 2 of a text block and a round key

AddRoundKey(L1,L2,n/L3) *** n – the number of a round

Q1 ⇒ Q3 n<2 ⇒ n
 L1.0 ⊕ L2n ⇒ L3.0
 Δn L1.1 ⊕ L2n ⇒ L3.1
 Δn L1.2 ⊕ L2n ⇒ L3.2
 Δn L1.3 ⊕ L2n ⇒ L3.3 **

Byte substitution with the help of S-box

SubBytes(L1,L4/L1)

*** every byte in L1 with the value *i* is substituted for the youngest byte
 *** of *i*th element in L4

~i FFh⇒m
 §1 Δi ⊕ Q1↔2
 L1i & m ⇒ a L1i > 8 & m ⇒ b L1i > 16 & m ⇒ c L1i > 24 ⇒ d
 L4d < 8 ∨ L4c < 8 ∨ L4b < 8 ∨ L4a ⇒ L1i →1
 §2 **

Cyclic shift of state rows

ShiftRows(L1/L1)

~i
 §1 Δi ⊕ Q1↔2
 i & 3 < 3 ⇒ n 32-n ⇒ t L1i > t ⇒ q L1i < n ∨ q ⇒ L1i →1
 §2 **

Mixing bytes in state columns

MixColumns(L1/L1)

@+L2(4) 4 ⇒ Q2 ~j
 §11 Δj ⊕ 4 ↔ 2 j < 3 ⇒ q ~k
 §111 Δk ⊕ 4 ↔ 112 L1k > q & FFh ⇒ L2k →111
 §112 *MixColumn(L2/L2) ~k
 §113 Δk ⊕ 4 ↔ 11 FFh < q ~ & L1k ⇒ L1k L2k < q ∨ L1k ⇒ L1k →113
 §2 **

Product of a vector-column and a matrix over the field GF(2⁸)

MixColumn(L1/L1)

@+L3(4) @+L4(4) Q1 ⇒ Q3 ⇒ Q4 ~i
 §1 Δi ⊕ Q3 ↔ 2 *xtime(L1i/a) a ⇒ L3i →1
 §2 L3.0 ⊕ L3.1 ⊕ L1.1 ⊕ L1.2 ⊕ L1.3 ⇒ L4.0
 L1.0 ⊕ L3.1 ⊕ L3.2 ⊕ L1.2 ⊕ L1.3 ⇒ L4.1
 L1.0 ⊕ L1.1 ⊕ L3.2 ⊕ L3.3 ⊕ L1.3 ⇒ L4.2
 L3.0 ⊕ L1.0 ⊕ L1.1 ⊕ L1.2 ⊕ L3.3 ⇒ L4.3 ~i
 §3 Δi ⊕ 4 ↔ 4 L4i ⇒ L1i →3
 §4 **

Multiplication of a field GF(2⁸) element by *x* (mod $x^8 + x^4 + x^3 + x + 1$)

xtime(x/x)

x < 1 ⇒ x & 100h ↔ 0 x ⊕ 11Bh ⇒ x
 §0 **

Key expansion

KeyExpansion(L5,L4/L2)

@+L3(12) *DefineVi(L3/L3) *** L3 – constants v_i in the expansion procedure
 $44 \Rightarrow Q2 \text{ } \bar{i} \text{ Or}$

§1 $\Delta i \oplus 4 \leftrightarrow 2 \text{ } L5i \Rightarrow L2i \text{ } \rightarrow 1$

§2 $\Delta r \oplus 11 \leftrightarrow 3 \text{ } r < 2 \Rightarrow j$

§21 $j-1 \Rightarrow s-3 \Rightarrow k \text{ } j > 2 \Rightarrow m \text{ } L2s > 24 \Rightarrow q \text{ } L2s < 8 \vee q \Rightarrow q$

*SubWord(q,L4/q) *** byte substitution in q according to S-box

$L2k \oplus q \oplus L3m \Rightarrow L2j \text{ } \Delta j \text{ } \bar{n}$

§22 $\Delta n \oplus 3 \leftrightarrow 2 \text{ } j-1 \Rightarrow s-3 \Rightarrow k \text{ } L2k \oplus L2s \Rightarrow L2j \text{ } \Delta j \text{ } \rightarrow 22$

§3 **

Computer experiments show that the encryption speed for russian text (L. N. Tolstoy. War and Piece) on a computer with the processor i5-2540M, 250 GHz, 4 GB memory is near to 2.4 MB/c.

BIBLIOGRAPHY

1. *Mollin R. A.* An Introduction to Cryptography. Boca Raton, London, New York: Chapman & Hall/CRC, 2007.
2. *Tokareva N. N.* Symmetric Cryptography. Short Course: text-book. Novosibirsk: NSU, 2012. (in Russian).
3. *Agibalov G. P., Lipsky V. B., and Pankratova I. A.* Cryptographic extension of Russian programming language // Applied Discrete Mathematics. Application. 2013. No. 6. P. 93–98.

Секция 6

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ
В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 519.688

О ВОЗМОЖНОСТИ СОКРАЩЕНИЯ ПЕРЕБОРА
В АЛГОРИТМЕ БАЛАША

Н. В. Анашкина

Предлагается оптимизация алгоритма Балаша на основании исследования особенностей геометрического строения окрестностей тупиковых точек.

Ключевые слова: алгоритм Балаша, невязка, тупиковая точка.

Широкий класс задач дискретной математики сводится к анализу и решению систем нелинейных уравнений. Известны методы сведения этих систем [1, 2] к системам линейных ограничений вида

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n \geq b_1, \\ \vdots \\ a_{m1}x_1 + \dots + a_{mn}x_n \geq b_m, \end{cases} \quad (1)$$

где a_{ij} , b_i — целые числа; $x_i \in \{0, 1\}$, $i = 1, \dots, m$, $j = 1, \dots, n$.

Для нахождения решения (1) можно использовать как методы, работающие в действительной области, так и методы, позволяющие искать целочисленное решение [2–4]. Одним из таких алгоритмов является алгоритм Балаша, который предназначен для решения задач целочисленного программирования с булевыми переменными [1, 3, 5].

Для изложения алгоритма определим невязку системы (1) формулой

$$\mu(\mathbf{x}) = \sum_{b_i - \sum_j a_{ij}x_j > 0} \rho(\mathbf{x}, L_i),$$

где $\mathbf{x} = (x_1, \dots, x_n)$; $\rho(\mathbf{x}, L_i)$ — расстояние от вершины n -мерного куба \mathbf{x} до плоскости L_i , задаваемой i -м уравнением системы. Невязка $\mu(\mathbf{x})$ характеризует близость точки \mathbf{x} к невыполненным неравенствам.

С геометрической точки зрения неравенства системы (1) задают разделяющие плоскости в n -мерном пространстве и введённая невязка представляет собой сумму расстояний от вершины \mathbf{x} до отсекающих её плоскостей.

Алгоритм Балаша является локальным детерминированным траекторным алгоритмом. В качестве начального приближения \mathbf{x}_0 берётся вектор $(0, \dots, 0)$. В процессе поиска решения опробуются ближайšie к текущему векторы, на них вычисляется значение невязки системы, и следующим текущим становится вектор, на котором значение невязки минимально. Процесс продолжается до тех пор, пока не получим решение (нулевую невязку) или не попадём в тупиковую точку. Тупиковая точка характеризуется тем, что опробование всех векторов, отличающихся от полученного на

предыдущем шаге одной ненулевой координатой, не приводит к уменьшению невязки. Существует несколько стратегий выхода из тупиковых точек, каждая из которых в конечном итоге приводит к увеличению числа опробуемых векторов и в худшем случае — к тотальному перебору.

Для некоторых классов систем вычислительная сложность работы алгоритма линейна [2, 3, 6]. При решении произвольной системы неравенств не всегда удаётся избежать попадания в тупиковую точку, в этом случае число итераций алгоритма зависит от количества встретившихся тупиковых точек. Естественно, что оптимизирующие модификации алгоритма должны тем или иным способом вести к уменьшению их количества. Одним из вариантов возможных модификаций является изменение критерия выбора приоритетного направления перемещения по вершинам n -мерного куба на основании учёта геометрических особенностей расположения плоскостей в окрестностях тупиковых точек.

Анализ причин попадания в тупиковые точки в процессе поиска решения с помощью алгоритма Балаша позволяет выявить в процессе работы алгоритма направления, продвижения в сторону которых приведёт к попаданию в тупиковую точку. Предположим, что на i -м шаге алгоритма при текущем векторе \mathbf{x}_i минимальная невязка достигается на векторе \mathbf{x} , получаемом изменением j -й координаты \mathbf{x}_i , и при этом вектор \mathbf{x} не является решением. Далее опробуются смежные с \mathbf{x} векторы $\mathbf{y}_1, \dots, \mathbf{y}_k$, полученные заменой одной из его нулевых координат на 1. Рассмотрим следующие случаи расположения некоторой плоскости и вершин $\mathbf{x}, \mathbf{y}_1, \dots, \mathbf{y}_k$:

- 1) плоскость отсекает вершину \mathbf{x} и $\mathbf{y}_1, \dots, \mathbf{y}_k$;
- 2) плоскость отсекает $\mathbf{y}_1, \dots, \mathbf{y}_k$.

Переход в вершину \mathbf{x} при выполнении одного из условий обязательно приведёт в тупиковую ситуацию. Поэтому вершина \mathbf{x} не может стать текущей, даже если невязка для неё минимальна при опробовании на предыдущем шаге всех непройденных и смежных с вершиной \mathbf{x}_i вершин. Из изложенного можно сделать вывод о том, что j -я координата вектора-решения равна нулю. Такой подход позволяет расставлять не только единицы, но и нули, строя приближение к вектору решений.

С алгоритмической точки зрения попадания в тупиковую ситуацию в ряде случаев можно избежать, анализируя состояние системы в момент перехода в следующую вершину. При этом достаточно ввести для опробуемых и текущей вершин векторы-индикаторы выполнившихся неравенств. С их помощью легко выявить возникновение в ходе работы алгоритма случаев 1 и 2. Модификация алгоритма требует дополнительной емкостной сложности $O(m)$ и вычислительной сложности $O(n)$.

Проиллюстрируем работу модифицированного алгоритма на примере решения следующей системы линейных ограничений:

$$\left\{ \begin{array}{l} x_1 + x_2 + 2x_3 + 5x_5 - 6x_6 + x_7 \geq -5, \\ -3x_1 - x_2 - 2x_3 - x_4 - 2x_5 - x_6 - x_7 \geq -3, \\ x_1 + 2x_2 + 3x_3 - 7x_4 + 5x_5 + 2x_6 + x_7 \geq -3, \\ -x_1 - 3x_2 + 4x_3 + 2x_4 + 6x_5 + 5x_6 + 8x_7 \geq 4, \\ x_1 + x_2 + 4x_3 - 3x_4 + 2x_5 + 5x_6 + 6x_7 \geq 3, \\ -x_1 + 7x_2 + x_3 + 8x_4 + 2x_5 + 3x_6 + 8x_7 \geq 12, \\ x_2 - x_3 + x_4 - 2x_5 + x_6 - 4x_7 \geq 0. \end{array} \right. \quad (2)$$

Традиционный алгоритм Балаша стартует из вершины $x_0 = (0, 0, 0, 0, 0, 0, 0)$, в которой невязка системы $\mu(x_0) = 19$. Опробование одной проставленной единицы даёт минимальное значение невязки, равное 8, на векторе $x_1 = (0, 0, 0, 0, 0, 0, 1)$.

Из этого вектора, в свою очередь, последовательно переходим в $x_2 = (0, \mathbf{1}, 0, 0, 0, 0, \mathbf{1})$ с $\mu(x_2) = 3$, $x_3 = (0, \mathbf{1}, 0, 0, 0, \mathbf{1}, \mathbf{1})$ с $\mu(x_3) = 2$, $x_4 = (0, \mathbf{1}, 0, \mathbf{1}, 0, 0, \mathbf{1})$ с $\mu(x_4) = 2$ и попадаем в тупиковую точку. Эта ветвь алгоритма потребовала проведения 25 вычислений невязок и так и не привела к успеху.

Предложенный модифицированный алгоритм уже на первом шаге устанавливает запрет на проставление единицы в первой и седьмой координатах вектора и присваивает шестой координате значение 1: $x_1 = 0$, $x_7 = 0$ и $x_6 = 1$. Отметим, что срабатывают первый и второй случаи стратегии запрета перехода в следующую вершину и блокируются возможности расстановки единиц соответственно по переменным x_7 и x_1 , при этом $\mu(0, 0, 0, 0, 0, 0, 1) = 8$ и является минимальной из всех посчитанных на первом шаге. В результате $x_1 = (0, 0, 0, 0, 0, \mathbf{1}, 0)$. На втором шаге запрещается присваивание единицы переменным x_3 и x_5 , переменной x_2 даётся значение 1 и получается вектор $x_2 = (0, \mathbf{1}, 0, 0, 0, \mathbf{1}, 0)$. Присваивание единицы четвёртой координате приводит к нахождению решения $x_3 = (0, 1, 0, 1, 0, 1, 0)$. При этом модифицированная версия алгоритма потребовала вычисления невязки всего 11 раз.

Экспериментальные исследования, которые проводились на случайных системах неравенств, показали, что эффективность применения как базового алгоритма Балаша, так и его модификации зависит от структуры системы неравенств. Основным достоинством предложенной модификации алгоритма Балаша является исключение попадания в целые классы тупиковых точек.

ЛИТЕРАТУРА

1. Балакин Г. В., Никонов В. Г. Методы сведения булевых уравнений к системам пороговых соотношений // Обзорение прикладной и промышленной математики. 1994. Т. 1. Вып. 3. С. 389–401.
2. Рыбников К. К., Никонов Н. В. Прикладные задачи, сводящиеся к анализу и решению систем линейных неравенств. Метод разделяющих плоскостей // Вестник Московского государственного университета леса — Лесной вестник. 2002. № 2(22). С. 191–195.
3. Анашкина Н. В. Использование алгоритма Балаша для нахождения решения системы линейных ограничений специального вида // Вестник Московского государственного университета леса — Лесной вестник. 2004. № 4(35). С. 176–179.
4. Кофман А., Анри-Лабордер А. Методы и модели исследования операций. М.: Мир, 1977. 432 с.
5. Анашкина Н. В. Обзор методов решения систем линейных неравенств // Вестник Московского государственного университета леса — Лесной вестник. 2004. № 1(32). С. 144–148.
6. Гришукин В. П. Среднее число итераций в алгоритме Балаша // Сб. статей. Численные методы в линейном программировании. М.: Наука, 1973. С. 31–38.

УДК 519.688

СРАВНИТЕЛЬНЫЙ АНАЛИЗ НЕКОТОРЫХ АЛГОРИТМОВ РАСПОЗНАВАНИЯ ГЛАДКИХ ЧИСЕЛ

Д. С. Арбузов, Л. И. Туктарова

Приводятся результаты экспериментальных исследований трёх алгоритмов нахождения чисел, разложимых в заданной факторной базе: просеивания (с делением и логарифмического) и Бернштейна.

Ключевые слова: гладкие числа, просеивание, алгоритм Бернштейна.

Пусть задана факторная база S — множество, состоящее из некоторых простых чисел.

Определение 1. Целое число будем называть S -гладким, если все его простые делители входят в S .

Задачу нахождения большого количества гладких чисел приходится решать в некоторых методах факторизации чисел (например, квадратичное решето, решето числового поля) и дискретного логарифмирования. Рассмотрим два метода её решения: просеивания (в двух вариантах) и Бернштейна.

1. Метод просеивания [1]

Задан многочлен $F(x) \in \mathbb{Z}[x]$, факторная база S и целые числа $A, C, A < C$. Требуется найти все такие $x, A \leq x \leq C$, что $F(x)$ — S -гладкое. Строится таблица, ячейки которой занумерованы целыми числами от A до C . Метод заключается в заполнении таблицы, её изменении и последующем просмотре.

Алгоритм 1. Метод просеивания

Вход: S — факторная база; A, C — целые числа, $A < C$; многочлен $F(x) \in \mathbb{Z}[x]$;

$M = \max_{A \leq x \leq C} |F(x)|$; первоначальное заполнение таблицы T :

x	A	$A+1$	\dots	C
$F(x)$	$F(A)$	$F(A+1)$	\dots	$F(C)$

Выход: множество чисел $x, A \leq x \leq C$, таких, что $F(x)$ — S -гладкое.

- 1: Для всех $q \in S, l \in \{1, 2, \dots, \ln M / \ln q\}$
 - 2: найти все решения x_0 сравнения $F(x) = 0 \pmod{q^l}$.
 - 3: Для всех решений x_0
 - 4: содержимое ячеек $T(x)$ с номерами $x = x_0 + hq^l, (A - x_0)/q^l \leq h \leq (C - x_0)/q^l$, поделить на q .
 - 5: Просмотреть содержимое всех ячеек. Значение $F(x)$ является S -гладким тогда и только тогда, когда $T(x) = \pm 1$.
-

В отличие от метода пробных делений, в данном алгоритме все деления выполняются точно, то есть нет «неудачных» делений.

Улучшение алгоритма 1 (логарифмический вариант) заключается в том, что вычисляется не $F(x)$, а $\log |F(x)|$; на шаге 4 вместо деления на q вычитается $\log q$; на шаге 5 значения $T(x)$ сравниваются с 0. В результате получаем выигрыш как по времени выполнения алгоритма, так и по требуемой памяти. Однако из-за погрешностей округления в вычислении логарифма даже для S -гладких значений $F(x)$ можем получить $T(x) \neq 0$, и шаг 5 надо модифицировать следующим образом: значение $F(x)$ является S -гладким тогда и только тогда, когда $|T(x)| \leq E$ для некоторого «малого» E . При этом возможны ошибки первого и второго рода: необнаружение гладкого числа (влияет на эффективность алгоритма) и принятие негладкого числа за гладкое (влияет на корректность алгоритма).

Для оценки величины E проведён следующий эксперимент: $F(x) = (x + m)^2 - N$, где $m = \sqrt{N}$, N — модуль RSA длиной $|N|$ бит; логарифм берётся по основанию 2 и округляется до ближайшего целого снизу; факторная база S состоит из $|S|$ первых простых чисел q , таких, что $\left(\frac{N}{q}\right) = 1$ (параметры соответствуют методу квадратичного решета для факторизации числа N). Результаты представлены в табл. 1. Здесь

Min — минимальное значение $T[x]$, полученное после просеивания и соответствующее негладкому числу; Max — максимальное значение $T[x]$, которое соответствует гладкому числу. Видно, что во всех случаях можно выбрать такое E , что $\text{Max} < E < \text{Min}$. Это позволит избежать ошибок обоих родов.

Таблица 1

Оценка параметра E

	$ S = 20$		$ S = 30$		$ S = 50$		$ S = 100$	
$ N $	Min	Max	Min	Max	Min	Max	Min	Max
20	7,6	3,2	8	2,8	9	3,2	10,6	3,2
40	8,8	4,2	8,6	4,6	9,6	4,6	11	7
50	8,8	6	9	5,2	9,8	6,6	10,8	5,8
60	10,2	5,4	10	6,3	10	6,4	11,2	7,4

Эксперименты показали, что логарифмический вариант просеивания работает быстрее, чем вариант с делением; результаты приведены в табл. 2 (CPU Intel Pentium P6100 2,00 ГГц, память 1,7 Гбайт, swap 1,9 Гбайт).

Таблица 2

Выигрыш логарифмического просеивания (по времени)

$ S $	$ N $	$C - A$	Выигрыш, %
30	50	2^{17}	41,16
50	50	2^{15}	55,2
50	60	2^{18}	48,72
100	50	2^{13}	45
100	60	2^{17}	45,24
150	50	2^{13}	56
150	60	2^{16}	43
150	70	2^{17}	49

Кроме того, хранение логарифмов чисел вместо самих чисел позволяет экономить используемую память. Так, при размере числа $|N| = 64$ бита, длине отрезка $C - A = 2^{18}$ и факторной базе из 50 элементов во время работы программы, реализующей просеивание с делением, используется память файла подкачки на 50% (972 Мбайт), в то время как программа с вычислением логарифмов использует лишь 27% (525 Мбайт). Это сильно влияет на время выполнения программы; в первом случае оно составляет 129 с, во втором — 42 с.

2. Метод Бернштейна [2]

Метод Бернштейна позволяет сократить объём вычислений при помощи использования деревьев. Задача формулируется так: заданы факторная база S и множество целых чисел P . Требуется найти все S -гладкие $p \in P$.

Отличие в постановке задачи от метода просеивания состоит в том, что множество чисел произвольно.

Обозначим $Q = \prod_{s \in S} s$ и найдём остатки от деления Q на числа $p \in P$. Пусть $Q \bmod p = r$. Тогда $r = ab$, где $a = (Q, p)$ — произведение тех простых, которые присутствуют в базе S и в исследуемом числе. Пусть наибольший показатель в каноническом разложении числа p на простые множители не превосходит z . Тогда в каноническом

разложении r^z все простые присутствуют в степени, не меньшей, чем в p . Таким образом, $(r^z \bmod p, p)$ есть часть числа p , разложимая в базе S ; и число p является S -гладким, если $(r^z \bmod p, p) = p$, т. е. $r^z \bmod p = 0$.

Определение 2. Деревом произведений для множества чисел называется двоичное дерево, листья которого соответствуют элементам этого множества и каждому узлу сопоставлено число, равное произведению чисел, соответствующих его потомкам.

Алгоритм 2. Метод Бернштейна

Вход: S — факторная база; множество чисел P ; $|P| \geq |S|$.

Выход: все S -гладкие числа $p \in P$.

- 1: Построить дерево произведений для факторной базы S . В корне этого дерева получим Q .
 - 2: Построить дерево произведений T для множества P (при этом вычисляются только значения, не большие Q , остальные помечаются *).
 - 3: Вычислить дерево остатков $Q \bmod T$ следующим образом: сначала число R , приписанное корню дерева T , заменяется на $Q \bmod R$, а затем происходит спуск по ветвям, во время которого каждое число заменяется на его остаток от деления на новое значение, приписанное его родителю.
 - 4: Найти наименьшее натуральное число k , для которого $\max P \leq 2^{2^k}$.
 - 5: Для всех $p \in P$:
 найти $r = Q \bmod p$ в дереве остатков $Q \bmod T$;
 вычислить $s_p := r^{2^k} \bmod p$.
 - 6: Ответ: все числа p , для которых $s_p = 0$.
-

Замечание. Значение z выбирается в виде 2^k для того, чтобы возведение в степень z выполнить за k возведений в квадрат.

В табл. 3 приведены результаты сравнения метода просеивания (с логарифмированием) и метода Бернштейна.

Таблица 3

Сравнение методов просеивания и Бернштейна

Метод	$ S = 50$				$ S = 100$			
	$ N = 50$		$ N = 60$		$ N = 50$		$ N = 60$	
	Время, с	Память, Мбайт	Время, с	Память, Мбайт	Время, с	Память, Мбайт	Время, с	Память, Мбайт
Просеивание	4	200	19	880	3	155	23	990
Метод Бернштейна	3	5	13	5	3	29	25	30

Таким образом, метод Бернштейна существенно выигрывает по памяти и времени при небольшой мощности факторной базы, однако при увеличении размеров факторной базы и просеиваемых чисел метод просеивания с логарифмированием догоняет и обгоняет метод Бернштейна по времени, хотя и существенно проигрывает по памяти.

ЛИТЕРАТУРА

1. Глухов М. М., Круглов И. А., Пижчур А. Б., Черемушкин А. В. Введение в теоретико-числовые методы криптографии: учебник для вузов. М.: Лань, 2011.
2. Крендалл Р., Померанс К. Простые числа: криптографические и вычислительные аспекты. М.: УРСС, Либроком, 2011.

УДК 519.7

О GPU-РЕАЛИЗАЦИИ ОГРАНИЧЕННОЙ ВЕРСИИ НЕХРОНОЛОГИЧЕСКОГО АЛГОРИТМА DPLL¹

В. Г. Булавинцев, А. А. Семенов

Описан новый решатель *ngsat*, предназначенный для решения SAT-задач на GPU (Graphic Processor Unit). Данный решатель основан на ограниченном варианте нехронологического DPLL без процедуры Clause Learning; в нём использованы специальные приемы, позволяющие повысить эффективность исполнения DPLL на SIMD-архитектуре. Приводятся результаты тестирования *ngsat* на задачах поиска систем ортогональных латинских квадратов.

Ключевые слова: GPU, алгоритм DPLL, SAT, параллельные вычислительные архитектуры, CUDA, SIMD.

Прошло всего несколько лет с момента, когда стало возможным использовать GPU для неграфических вычислений. За это время на GPU перенесено множество алгоритмов из различных отраслей науки и техники. Этот опыт показал, что класс алгоритмов, допускающих успешную реализацию на GPU, к сожалению, весьма узок. К примеру, в криптографии GPU применяются, главным образом, для реализации различных brute force-атак [1]. Серьёзные проблемы возникают при переносе на GPU комбинаторных алгоритмов, в которых используются различные стратегии обхода дерева поиска. Основным негативным эффектом в этом плане является так называемый «SIMD-эффект», когда при выполнении условного перехода часть вычислительных ядер SIMD-группы вынуждена простаивать в ожидании необходимой команды.

Одной из актуальных является проблема реализации на GPU современных алгоритмов решения SAT-задач [2] как алгоритмов, применимых в различных разделах вычислительной дискретной математики. Мы представляем реализованный на GPU SAT-решатель, названный *ngsat*. В данном решателе использован ряд специально разработанных техник и алгоритмов. В частности, используются специальные структуры данных для представления дизъюнктов. Эти структуры несколько медленнее стандартных «watched literals» [3], однако существенно более экономны по памяти, что даёт возможность запускать на GPU параллельно большое число SAT-задач. В решателе *ngsat* существенно снижена негативная составляющая SIMD-эффекта за счёт применения специальных процедур голосования и реорганизации циклов. Кратко, техника голосования заключается в периодическом опросе работающих в одной SIMD-группе вычислительных ядер о том, какую команду они готовы исполнять. По результатам такого опроса контроллер подает на SIMD-группу команду, к выполнению которой готово большинство ядер. Техника реорганизации циклов также используется для нивелирования SIMD-эффекта и заключается в перестройке вложенных циклов в процедуре «Unit Propagation». Процесс обработки очереди в «Unit Propagation» состоит из двух стадий — выбора переменной из очереди и обработки дизъюнктов, инцидентных выбранной переменной. Может оказаться так, что в некоторый момент одно ядро SIMD-группы обработало все дизъюнкты некоторой переменной и вынуждено ждать окончания аналогичного действия другого ядра. Показано, что две эти стадии в «Unit Propagation» можно очень просто перестроить таким образом, что такого простоя не возникнет.

¹Работа выполнена при поддержке гранта РФФИ № 11-07-00377 а.

В решателе `ngsat` реализована версия DPLL с ограниченной формой нехронологического бэктрекинга. Хорошо известно, что нехронологический бэктрекинг существенно эффективнее обычного (хронологического). Однако реализации полноценного нехронологического DPLL мешает небольшой (в пересчёте на одну задачу) объём памяти GPU. Исходя из этого, в `ngsat` реализован ограниченный вариант нехронологического DPLL, общая идея которого описана в различных источниках (см., например, [4]). Эта идея состоит в том, чтобы для каждой переменной, значение которой выведено по ВСП, хранить информацию о породивших этот вывод причинах. Для этой цели можно использовать стандартные приёмы, применяемые во всех современных SAT-решателях [5]. Хранения конфликтных дизъюнктов можно избежать, если в процессе вывода не использовать рестарты. Данный факт ограничивает «мощность вывода» [6], однако для целого ряда задач такой подход вполне оправдан.

Решатель `ngsat` протестирован на SAT-задачах, кодирующих поиск систем ортогональных латинских квадратов. Эти задачи являются очень трудными [7]. На данном классе тестов `ngsat` оказался более эффективным, чем широко известный SAT-решатель `minisat` [8].

ЛИТЕРАТУРА

1. Беспалов Д. В., Булавинцев В. Г., Семенов А. А. Использование графических ускорителей в решении задач криптоанализа // Прикладная дискретная математика. Приложение. 2010. №3. С. 86–87.
2. Biere A., Heule M., van Maaren H., and Walsh T. (eds.). Handbook of satisfiability. IOS Press, 2009.
3. Een N. and Sorensson N. An extensible SAT-solver // LNCS. 2003. V. 2919. P. 502–518.
4. Marques-Silva J. P. and Sakallah K. A. GRASP: A search algorithm for propositional satisfiability // IEEE Trans. Comp. 1999. V. 48. No. 5. P. 506–521.
5. Moskewicz M. W. et al. Chaff: Engineering an efficient SAT solver // Proc. 38th Design Automation Conference. Las Vegas, NV, USA: ACM, 2001. P. 530–535.
6. Beame P., Kautz H. A., and Sabharwal A. Towards understanding and harnessing the potential of clause learning // J. Artif. Intell. Res. (JAIR). 2004. V. 22. P. 319–351.
7. Zhang H. Combinatorial designs by SAT solvers // Handbook of satisfiability. IOS Press, 2009. P. 533–568.
8. <http://minisat.se/>

УДК 519.712

ОБ АСИМПТОТИКЕ РЕШЕНИЙ РЕКУРРЕНТНЫХ СООТНОШЕНИЙ В АНАЛИЗЕ АЛГОРИТМОВ РАСЩЕПЛЕНИЯ ДЛЯ ПРОПОЗИЦИОНАЛЬНОЙ ВЫПОЛНИМОСТИ

В. В. Быкова

Исследована традиционная техника анализа алгоритмов расщепления для решения задачи пропозициональной выполнимости. Предложена теорема, устанавливающая асимптотические верхние оценки времени работы алгоритмов в случае сбалансированного расщепления.

Ключевые слова: алгоритмы расщепления, сложность вычислений.

Задача пропозициональной выполнимости (Satisfiability, SAT) является одной из известных NP-полных задач дискретной математики. Пусть X — множество булевых

переменных, т. е. переменных, принимающих значения `true` и `false`. Булева формула в конъюнктивной нормальной форме (КНФ) представляет собой конъюнкцию конечного числа клозов, где клоз — дизъюнкция конечного числа литералов, не содержащая ни одной переменной одновременно с её отрицанием, а всякий литерал — некоторая переменная $x \in X$ или её отрицание $\neg x$. Длина клоза определяется как количество его литералов, а длина формулы — как сумма длин её клозов. Задача SAT традиционно формулируется следующим образом: для булевой формулы F в КНФ выдать ответ «Выполнима», если существует выполняющий набор — набор значений переменных, при котором каждый клоз формулы F принимает значение `true`, и в противном случае выдать ответ «Невыполнима».

Переборный алгоритм решения задачи выполнимости с $N = |X|$ переменными сводится к анализу 2^N различных наборов этих переменных и определяет тривиальную верхнюю оценку сложности для SAT в худшем случае. Алгоритмы, направленные на улучшение тривиальной верхней оценки, активно разрабатываются и исследуются с начала 60-х годов прошлого века и до настоящего времени. Многие из них основаны на принципе «разделяй и властвуй» и названы расщепляющими алгоритмами [1]. Алгоритмы, приведённые в работах М. Дэвиса и Х. Патмена [2] и М. Дэвиса, Г. Лоджмана и Д. Лавленда [3], считаются первыми расщепляющими алгоритмами для задачи SAT и называются DPLL-алгоритмами (по первым буквам фамилий всех четырёх авторов). Большинство расщепляющих алгоритмов, разработанных за последние пятьдесят лет для SAT, основаны на идеях, заложенных в DPLL-алгоритмах. Такой алгоритм сначала упрощает входную формулу, после чего расщепляет полученную формулу и производит рекурсивные вызовы самого себя для формул меньшей сложности. Мерами сложности формул обычно выступают число переменных $N = |X|$, число клозов K или длина формулы $L = |F|$. Процесс расщепления в общем виде можно подставить следующим образом.

Вход: формула F в КНФ.

- 1) Редуцировать F , т. е. преобразовать формулу F в формулу F_0 меньшей сложности, применяя некоторый конечный набор правил редукции.
- 2) Если задача SAT тривиально решается для F_0 , то выдать соответствующий ответ.
- 3) Выбрать переменную x , входящую в F_0 , используя определенную эвристику.
- 4) Выполнить расщепление формулы F_0 на конечное число формул меньшей сложности по некоторому правилу, использующему различные значения переменной x . Осуществить рекурсивные вызовы данного алгоритма для этих формул. Если хотя бы один из рекурсивных вызовов возвратил выполняющий набор, скорректировать его надлежащим образом и вернуть результат. В противном случае выдать ответ «Невыполнима».

Естественные требования к алгоритму расщепления: редуцирование F , эвристический выбор переменной x и построение решения для F из решения для F_0 должны выполняться за полиномиальное время относительно выбранной меры сложности формул. Расщепляющие алгоритмы, как правило, различаются набором правил редукции, эвристикой выбора переменной x , мерой сложности формул и правилом расщепления.

В настоящее время известно большое количество правил редукции и эвристик. Наиболее популярные правила редукции можно найти в [1–5]. Типичными эвристиками по выбору переменной для расщепления являются «взять любую переменную из самого короткого клоза» или «предпочтение отдать переменной, входящей в наибольшее чис-

ло кловов». Простейшее правило расщепления сводится к замене на шаге 4 формулы F_0 на две формулы $F_1 = F_0[x]$ и $F_2 = F_0[\neg x]$. При этом формула F_1 получается из формулы F_0 путём присваивания значения **true** переменной x , что влечёт удаление всех кловов, содержащих x без отрицания, и удаление литерала $\neg x$ в оставшихся кловах. Аналогичным образом определяется формула F_2 , т. е. путём присваивания в F_0 переменной x значения **false**. Мера сложности формул F_1 и F_2 понижается по отношению к F за счет применения на шаге 1 правил редукции и присваивания значений переменной x на шаге 4. Таким образом, при использовании простейшего правила расщепления выполняются два рекурсивных вызова для формул F_1 и F_2 соответственно. Очевидно, что число рекурсивных вызовов увеличивается для расщеплений более сложного вида, к примеру, таких: $F_1 = F_0[x, y]$, $F_2 = F_0[x, \neg y]$, $F_3 = F_0[\neg x]$ или $F_1 = F_0[x, y]$, $F_2 = F_0[x, \neg y]$, $F_3 = F_0[\neg x, y]$, $F_4 = F_0[\neg x, \neg y]$.

Область применения алгоритмов расщепления не ограничивается задачей SAT. Данный класс алгоритмов плодотворно применяется при решении многих NP-полных задач (например, MAX-SAT, MAX-2-SAT, максимальное сечение и др.). Для задачи SAT имеется множество работ, описывающих алгоритмы расщепления, где каждая следующая работа улучшает результат предыдущей за счет введения новых правил редукции и расщепления, новой меры сложности формул или более тщательного анализа алгоритма с получением верхней оценки времени его работы. На алгоритмах расщепления базируются многие современные SAT-солверы.

Пусть n — некоторая мера сложности входной формулы F (например, число переменных или число кловов). Предположим, что на шаге 4 алгоритм расщепления разбивает формулу F_0 на m формул F_1, F_2, \dots, F_m сложности n_1, n_2, \dots, n_m соответственно и $1 \leq n_1 \leq n_2 \leq \dots \leq n_m < n$. Пусть временные затраты на редуцирование F , эвристический выбор переменной x и построение решения для F из решений F_1, F_2, \dots, F_m составляют $f(n)$, где $f(n)$ является неубывающей функцией субполиномиального или полиномиального порядка роста [6]. Если $T(n)$ обозначает время работы алгоритма расщепления для входной формулы F , то $T(n)$ является решением следующего рекуррентного соотношения:

$$T(n) = T(n_1) + T(n_2) + \dots + T(n_m) + f(n)$$

или

$$T(n) = T(n - t_1) + T(n - t_2) + \dots + T(n - t_m) + f(n), \quad (1)$$

где $t_1 = n - n_1, t_2 = n - n_2, \dots, t_m = n - n_m$. Вектор (t_1, t_2, \dots, t_m) принято называть вектором расщепления. Поскольку в векторе расщепления возможны равные элементы, то рекуррентное соотношение (1) можно преобразовать к неоднородному линейному рекуррентному соотношению с постоянными коэффициентами

$$T(n) = a_1 T(n - 1) + a_2 T(n - 2) + \dots + a_k T(n - k) + f(n) \quad (2)$$

и k начальными условиями $T(0) = \Theta(1), T(1) = \Theta(1), \dots, T(k - 1) = \Theta(1)$. В соотношении (2) величины a_1, a_2, \dots, a_k — целые константы, $a_1, a_2, \dots, a_{k-1} \geq 0, 1 \leq a_k \leq m, 1 \leq k \leq n$. Начальные условия свидетельствуют о том, что на формулах сложности k и менее время работы алгоритма расщепления ограничено сверху и снизу константой. Общих методов, дающих решение соотношения (2) в замкнутом виде, не известно. Когда $f(n) \equiv 0$, соотношение (2) становится однородным. Такие соотношения решаются с помощью нахождения корней соответствующего характеристического многочлена [7]. Известны некоторые приёмы учета неоднородности в (2).

Часто для нахождения асимптотической оценки решения рекуррентного соотношения (2) применяют дерево рекурсии [8]. Вычисление сводится к взвешиванию определённым образом вершин этого дерева и подсчёту числа вершин. Именно на таком подходе основана техника анализа алгоритмов расщепления, предложенная О. Кульманом и Х. Люкхардтом [4, 5]. Многие оценки для задачи SAT и родственных с ней NP-полных задач получены с помощью этой техники; она подробно описана в [1]. Известны программные реализации этой техники [9].

В работе детально исследована техника Кульмана и Люкхардта. Отмечены её достоинства и недостатки. Предложен и доказан новый вариант теоремы из работы [10]. Данная теорема позволяет находить непосредственно асимптотические верхние оценки для частного случая рекуррентного соотношения (2), когда выполняется сбалансированное расщепление (при $a_1 = a_2 = \dots = a_{k-1} = 0$, $a_k > 1$). Показана область применения этой теоремы для анализа алгоритмов расщепления. Теорема формулируется следующим образом.

Теорема 1. Пусть дано рекуррентное соотношение

$$T(n) = \begin{cases} \Theta(1), & \text{если } 0 \leq n \leq k-1, \\ aT(n-k) + f(n), & \text{если } n \geq k, \end{cases} \quad (3)$$

где $a > 1$, $k \geq 1$ — целые константы. Пусть $\tau \geq 0$ — вещественная константа. Тогда при $n \rightarrow \infty$

$$\begin{aligned} T(n) &= O(n^\tau a^{n/k}), & \text{если } f(n) &= O(n^\tau), \\ T(n) &= O(a^{n/k}), & \text{если } f(n) &\equiv 0. \end{aligned}$$

В рекуррентном соотношении (3) константа a трактуется как число формул, получаемых на шаге расщепления, при этом все формулы имеют одну и ту же сложность $(n-k)$. Вторая оценка данной теоремы отвечает случаю «бесплатного» рекурсивного перехода в алгоритмах расщепления.

ЛИТЕРАТУРА

1. *Всемирнов М. А., Гурш Э. А., Данцин Е. Я., Иванов С. В.* Алгоритмы для пропозициональной выполнимости и верхние оценки их сложности // Теория сложности вычислений. VI. Зап. научн. сем. ПОМИ. СПб., 2001. Т. 277. С. 14–46.
2. *Davis M. and Putman H.* A computing procedure for quantification theory // J. ACM. 1960. No. 7(3). P. 201–215.
3. *Davis M., Logemann G., and Loveland D.* A machine program for theorem-proving // Comm. ACM. 1962. No. 5(7). P. 394–397.
4. *Kullmann O.* New methods for 3-SAT decision and worst-case analysis // Theor. Comp. Sci. 1999. No. 223. P. 1–72.
5. *Kullmann O. and Luckhardt H.* Deciding propositional tautologies: Algorithms and their complexity / Preprint. 1997. <http://cs-svr1.swan.ac.uk/~csoliver>
6. *Быкова В. В.* Эластичность алгоритмов // Прикладная дискретная математика. 2010. № 2(8). С. 87–95.
7. *Грэхем Р., Кнут Д., Паташник О.* Конкретная математика. Основание информатики. М.: Бином. Лаборатория знаний, 2006.
8. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ. М.: МЦНМО, 1999.

9. Куликов А. С., Федин С. С. Автоматические доказательства верхних оценок на время работы алгоритмов расщепления // Теория сложности вычислений. IX. Зап. научн. сем. ПОМИ. СПб., 2004. Т. 316. С. 111–128.
10. Быкова В. В. Математические методы анализа рекурсивных алгоритмов // Журнал Сибирского федерального университета. Сер. Математика и физика. 2008. № 1(3). С. 236–246.

УДК 519.6

К РЕШЕНИЮ БОЛЬШИХ СИСТЕМ СРАВНЕНИЙ

К. Д. Жуков, А. С. Рыбаков

Пусть дано конечное множество S натуральных чисел, такое, что почти все его элементы попарно взаимно просты. Рассматривается алгоритм нахождения всех элементов $s \in S$, таких, что $(s, s') > 1$ для некоторого $s' \in S$, $s' \neq s$, который позволяет сводить произвольную систему полиномиальных сравнений к нескольким системам с взаимно простыми модулями.

Ключевые слова: взаимно простая база, наибольший общий делитель, дерево наибольших общих делителей, НОД слиянием.

Определение 1. Пусть S — конечное подмножество натуральных чисел. Взаимно простой базой для S называется конечное подмножество натуральных чисел B , такое, что любое число из S представляется в виде произведения элементов из B и любые два элемента множества B взаимно просты.

Взаимно простые базы используются в ряде приложений, например при решении систем сравнений в целых числах [1].

Пусть задана система полиномиальных сравнений вида $f_i(x) = 0 \pmod{a_i}$, $i = 0, 1, \dots, m - 1$, где x — набор переменных. Построим для множества чисел $\{a_0, \dots, a_{m-1}\}$ взаимно простую базу $\{b_0, \dots, b_{r-1}\}$. Для каждого элемента b_j нужно последовательно составить системы сравнений по модулям b_j^l , где $l = 1, 2, \dots, t_j$, а t_j — максимальное, для которого найдётся a_i , делящееся на $b_j^{t_j}$. В эти системы войдут те уравнения $f_i(x) = 0$, для которых число a_i делится на $b_j^{t_j}$. Поднимая по лемме Гензеля решения от системы к системе, получим множество решений по модулю $b_j^{t_j}$. Найдя такие решения для каждого b_j , применим китайскую теорему об остатках, чтобы получить решения вида $x \pmod{\prod_j b_j^{t_j}}$. Каждое такое решение будет решением исходной системы.

Построение взаимно простой базы применяется и в других приложениях, например в задачах нахождения алгебраических зависимостей среди радикалов [2], вычисления нормальных базисов в расширениях полей [3] и др. [4].

В работе [4] Д. Бернштейн предложил алгоритм построения взаимно простой базы с некоторыми дополнительными свойствами, называемой естественной взаимно простой базой.

Ниже предлагается подход к построению взаимно простой базы для специального случая. Подход эффективен, когда заранее известно, что нетривиальная часть взаимно простой базы (элементы, отличные от 1 и не содержащиеся в S) достаточно мала. Метод состоит в том, чтобы быстро отбросить элементы $s \in S$, такие, что $(s, s') = 1$ для любого $s' \in S$, $s' \neq s$. Для остальных чисел можно применить известные алгоритмы построения взаимно простой базы.

С помощью известного алгоритма вычисления НОД слиянием можно проверить, являются ли все элементы множества S взаимно простыми. Данный алгоритм заключается в вычислении наибольших общих делителей в дереве произведений элементов множества S . С помощью небольшой модификации можно получить алгоритм 1 для выявления элементов $s \in S$, имеющих нетривиальный НОД с некоторым элементом из $S \setminus \{s\}$. Везде далее через \lg обозначен логарифм по основанию 2.

Алгоритм 1

Вход: $S = \{a_0, \dots, a_{m-1}\}$ — множество различных n -битовых натуральных чисел (для упрощения m — степень двойки)

Выход: T — подмножество S , такое, что для любого $s \in T$ и для некоторого $s' \in S$, $s' \neq s$, выполняется условие $(s, s') > 1$.

1: Для всех $i = 0, \dots, m - 1$

$a'_i \leftarrow a_i$.

2: Для всех $i = 1, 2, \dots, \lg m$

3: Для всех $j = 0, 1, \dots, 2^{\lg m - i} - 1$

4: $d \leftarrow (a'_{2j}, a'_{2j+1})$.

5: Если $d \neq 1$, то

6: Для всех $l = j2^i, j2^i + 1, \dots, (j + 1)2^i - 1$

7: Если $(d, a_l) \neq 1$, то

$T \leftarrow T \cup \{a_l\}$;

8: $a'_j \leftarrow a'_{2j} \cdot a'_{2j+1}$.

9: Вывести T

Важное свойство алгоритма 1 заключается в том, что в памяти нужно хранить только текущий уровень дерева, а не всё дерево. Отсюда следует, что алгоритму 1 требуется $O(mn)$ битов памяти. Трудоёмкость алгоритма складывается из трудоёмкости вычисления дерева наибольших общих делителей ($O(mn \lg m \lg^2(n\sqrt{m}) \lg \lg(mn))$ битовых операций) и трудоёмкости всех вызовов цикла в строках 6 и 7 ($O(kmn \lg m \times \lg(kn) \lg \lg(kn))$ битовых операций, где $k = |T|$).

Несложно видеть, что при $k < \lg^2(mn)$ трудоёмкость описанного алгоритма не превосходит оценки трудоёмкости алгоритма Бернштейна в $O(mn \lg m \lg^3(nm) \lg \lg(mn))$ битовых операций.

Известен ещё один подход к выявлению множества T , вытекающий из работы [5].

Вычисляется произведение $U = \prod_{i=0}^{m-1} a_i$ с помощью дерева произведений. Затем вычисляются величины $b_i = U \bmod a_i^2$, $i = 0, \dots, m - 1$, с помощью дерева остатков. На выход подаются те числа a_i , для которых $(b_i/a_i, a_i) \neq 1$. Трудоёмкость построения деревьев произведений и остатков оценивается в $O(mn \lg m \lg(nm) \lg \lg(mn))$ битовых операций. При этом построение дерева остатков требует $O(mn \lg m)$ битов памяти, что в $\lg m$ раз больше, чем требуемая память для алгоритма 1.

ЛИТЕРАТУРА

1. Bach E., Miller G, and Shallit J. Factor refinement // J. Algorithms. 1993. No. 15. P. 199–222.
2. Smedley T. J. Detecting algebraic dependencies between unnested radicals: extended abstract // Proc. Intern. Symp. on Symbolic and Algebraic Computation. Tokyo, Japan, 1990. P. 292–293.
3. Lunenburg H. On a little but useful algorithm // ААЕСС'85. LNCS. 1986. V. 229. P. 296–301.

4. Bernstein D. J. Factoring into coprimes in essentially linear time // J. Algorithms. 2005. No. 54(1). P. 1–30.
5. Bernstein D. J. Scaled reminder trees // <http://cr.yp.to/papers.html#scaledmod>

УДК 519.688

ОПТИМИЗАЦИЯ $(p - 1)$ -АЛГОРИТМА ПОЛЛАРДА

А. С. Климина

Приведены критерии выбора параметров $(p - 1)$ -алгоритма Полларда и рассмотрен метод его оптимизации.

Ключевые слова: $(p - 1)$ -алгоритм Полларда, факторизация чисел.

Рассмотрим $(p - 1)$ -алгоритм Полларда факторизации числа N [1]. Алгоритм состоит из следующих шагов.

Шаг 1. Выбираем число k .

Шаг 2. Выбираем произвольное a , $1 < a < N$.

Шаг 3. Вычисляем $d = (a, N)$. Если $d > 1$, получили нетривиальный делитель N .

Шаг 4. Вычисляем $D = (a^k - 1, N)$. Если $1 < D < N$, то D — нетривиальный делитель N . Если $D = 1$, возвращаемся к шагу 1, а при $D = N$ — к шагу 2.

Вопрос оптимального выбора параметра k не исследован до настоящего времени с нужной полнотой. Автором реализованы следующие подходы к выбору числа k : k — произведение нескольких случайных чисел; k — факториал некоторого числа, k — произведение степеней простых чисел; k — наименьшее общее кратное нескольких чисел. После анализа работы программы выбран третий метод, поскольку он работает быстрее остальных и даёт больше удачных разложений [2].

Зафиксируем $a = 2$ и рассмотрим работу $(p - 1)$ -алгоритма Полларда при следующих допущениях:

- 1) k — произведение нескольких первых простых чисел в заданной степени;
- 2) исследуемое число N представляет собой произведение двух простых множителей, p и q .

Пусть $D = (a^k - 1, N)$; $O_p(a)$, $O_q(a)$ — показатели числа a по модулям p и q соответственно. Возможны три случая:

- 1) k кратно ровно одному из чисел $O_p(a)$, $O_q(a)$. В этом случае получим $1 < D < N$, и нетривиальный делитель D найден;
- 2) k не кратно ни одному из $O_p(a)$, $O_q(a)$. В этом случае $D = 1$;
- 3) k кратно и $O_p(a)$, и $O_q(a)$. В этом случае $D = N$.

Таким образом, для удачного нахождения делителя N нужно выбрать параметр k не слишком большим и не слишком маленьким. Заметим, что k кратно $O_p(a)$ для любого a , если k кратно $p - 1$. Это, в свою очередь, гарантированно выполняется, если k является произведением всех простых чисел $p \leq \sqrt{N}/2$ в степенях $\log_p(\sqrt{N}/2)$. Такой выбор k хорошо работает для сравнительно небольших чисел N , однако при увеличении числа N становится неприемлемым.

Предлагается следующий алгоритм выбора k в процессе работы программы.

Пусть $k = p_1^\alpha p_2^\alpha \dots p_l^\alpha$, где p_1, \dots, p_l — первые l простых, α — некоторая константа. Будем рассматривать изменение k только за счёт изменения количества простых l . Идея выбора l состоит в том, чтобы выбрать его не слишком большим и не слишком малым, исходя из результатов работы программы (при слишком малом значе-

нии l результатом является 1, а при слишком большом — N). При этом l выбирается методом, похожим на половинное деление. При каждом значении l вычисляется $D(l) = (a^k - 1, N)$, $k = p_1^\alpha p_2^\alpha \dots p_i^\alpha$. Параметр l выбирается следующим образом:

Шаг 1. $l_{\min} := 1, l_{\max} := 1$.

Шаг 2. $l_{\max} := l_{\max} \cdot 2; l = l_{\max}$.

Шаг 3. Вычислить $D = D(l)$;

— если $D = 1$, то перейти к шагу 2;

— если $D = N$, то перейти к шагу 4;

— если $1 < D < N$ — выход, делитель найден.

Шаг 4. $l_{\min} := l_{\max}/2; l_{\text{midl}} := l_{\min} + (l_{\max} - l_{\min})/2; l := l_{\text{midl}}$.

Шаг 5. Вычислить $D = D(l)$;

— если $D = 1$, то $l_{\min} := l_{\text{midl}}; l_{\text{midl}} := l_{\min} + (l_{\max} - l_{\min})/2; l := l_{\text{midl}}$; перейти к шагу 5;

— если $D = N$, то $l_{\max} := l_{\text{midl}}; l_{\text{midl}} := l_{\min} + (l_{\max} - l_{\min})/2; l := l_{\text{midl}}$; перейти к шагу 5;

— если $1 < D < N$ — выход, делитель найден.

Проведены эксперименты на заранее сформированных массивах чисел, представляющих собой произведение двух простых, с разрядностью 20, 40 и 60 десятичных знаков. Из 10000 чисел разрядностью 20 десятичных знаков были разложены 4250, среднее время на обработку одного числа составило 1,42 с. Разрядность максимального числа, которое было разложено на множители, составляет 60 десятичных знаков:

$$\begin{aligned} & 1052808008400417645876606027989867285487720871343057551778083 = \\ & = 123547896523698521452369860571 \cdot 8521456358412963587456325968473, \end{aligned}$$

время разложения — 29 мин 23 с.

ЛИТЕРАТУРА

1. *Маховенко Е. Б.* Теоретико-числовые алгоритмы в криптографии. М.: Гелиос АРВ, 2006.
2. *Климина А. С.* Оптимизация выбора параметров для алгоритма Полларда // IV ОМНТК «Молодежь. Техника. Космос». СПб.: БГТУ, 2012. С. 285–286.

УДК 519.688

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ВЫЧИСЛЕНИЯ ФУНКЦИЙ РОСТА В КОНЕЧНЫХ ДВУПОРОЖДЁННЫХ ГРУППАХ ПЕРИОДА 5

А. С. Кузнецова, А. А. Кузнецов, К. В. Сафонов

Представлена параллельная версия алгоритма для вычисления функций роста в конечных двупорождённых группах периода 5.

Ключевые слова: *функция роста группы, диаметр Кэли, параллельный алгоритм.*

Пусть p — простое число, G — конечная группа экспоненты p . Это значит, что $g^p = e$ для всех $g \in G$. Так как G нильпотентна, то можно найти цепочку подгрупп $G = G_1 \supset \supset G_2 \supset \dots \supset G_n \supset G_{n+1} = e$, в которой G_i нормальны в G , а факторы G_i/G_{i+1} имеют порядок p и лежат в центре G/G_{i+1} .

Пусть для $1 \leq i \leq n$ элемент $a_i \in G_i$, но $a_i \notin G_{i+1}$, тогда каждый элемент группы $g \in G$ можно однозначным образом записать в виде

$$g = a_1^{\gamma_1} a_2^{\gamma_2} \dots a_n^{\gamma_n}, \quad 0 \leq \gamma_i < p. \quad (1)$$

Такое представление элементов группы (рс-представление) можно получить при помощи алгоритма, известного как p -quotient algorithm [1]. Он реализован в системах компьютерной алгебры GAP и Magma. Если \mathbb{A} — порождающее множество группы G , то любой её элемент, записанный в виде слова $\alpha_1\alpha_2\dots\alpha_s$, где $\alpha_i \in \mathbb{A}$, можно преобразовать к виду (1)

$$\alpha_1\alpha_2\dots\alpha_s \xrightarrow{pq} a_1^{\gamma_1}a_2^{\gamma_2}\dots a_n^{\gamma_n}. \quad (2)$$

Процедура (2) даёт возможность решить проблему равенства слов в G . Для вычисления функции роста и диаметра Кэли группы G относительно \mathbb{A} необходимо перечислить все элементы группы в формате минимальных слов [2]. Вычислив количество слов на каждой длине, можно получить функцию роста группы, а максимально возможная длина минимальных слов является диаметром Кэли группы.

Обозначим через $K_s(G)$ множество всех минимальных слов группы G , не превосходящих по длине s ; множество $Q_s(G)$ — элементы $K_s(G)$, записанные в виде правой части (2).

Пусть $s_0 \in \mathbb{N}$ — минимальное число, для которого выполняется $K_{s_0}(G) = K_{s_0+1}(G)$. В этом случае s_0 — диаметр Кэли группы G .

Опишем последовательный алгоритм, вычисляющий функцию роста группы G относительно $\mathbb{A} = \{a_1, a_2, \dots, a_m\}$:

- 1) $s = 0$, $K_0 = \{e\}$, $Q_0 = \{a_1^0 a_2^0 \dots a_n^0\}$, $T = K_0$.
- 2) $s = s + 1$, $K_s = K_{s-1}$, $Q_s = Q_{s-1}$, $V = a_1 T \cup a_2 T \dots \cup a_m T$, $T = \emptyset$, $i = 1$.
- 3) Для $v_i \in V$ $v_i \xrightarrow{pq} \hat{v}_i$. Если $\hat{v}_i \notin Q_s$, то $K_s = K_s \cup \{v_i\}$, $Q_s = Q_s \cup \{\hat{v}_i\}$, $T = T \cup \{v_i\}$.
- 4) Если $i < |V|$, то $i = i + 1$, переход в п. 3, иначе — в п. 5.
- 5) Если $T \neq \emptyset$, то переход в п. 2; иначе — в п. 6.
- 6) Диаметр G равен $s - 1$, $K_{s-1}(G)$ — множество всех минимальных слов группы. Функция роста задаётся формулой $f(j) = |K_j(G)| - |K_{j-1}(G)|$, $1 \leq j \leq s - 1$.

Завершение работы алгоритма.

Для увеличения скорости вычислений алгоритм можно распараллелить следующим образом. Множество Q_s разбивается на p^r непересекающихся классов элементов, где каждый класс однозначно определяется фиксированным набором значений $\gamma_1, \gamma_2, \dots, \gamma_r$. Параметр r определяется экспериментально и зависит от характеристик мультипроцессорной вычислительной системы.

Пусть $B_0(2, 5, k)$ — максимальная конечная двупорождённая бернсайдова группа периода 5 степени нильпотентности k . В данном классе групп наибольшей является группа $B_0(2, 5, 12)$, порядок которой равен 5^{34} . Для каждой из $B_0(2, 5, k)$ известны рс-представления, которые несложно получить, используя систему компьютерной алгебры GAP. Положим $\mathbb{A} = \{a_1, a_2\}$ — порождающие элементы $B_0(2, 5, k)$. Для $k \leq 6$ к настоящему времени вычислены функции роста и диаметры Кэли групп $B_0(2, 5, k)$ относительно \mathbb{A} . Отметим, что для нахождения функций роста при $k \geq 2$ использовались компьютерные вычисления. В таблице приведены значения порядков групп $B_0(2, 5, k)$, а также их диаметры $D_k(\mathbb{A})$ относительно \mathbb{A} .

k	$ B_0(2, 5, k) $	$D_k(\mathbb{A})$	k	$ B_0(2, 5, k) $	$D_k(\mathbb{A})$
1	5^2	8	4	5^8	30
2	5^3	10	5	5^{10}	32
3	5^5	20	6	5^{14}	45

Следующий нерешённый случай — $k = 7$. Поскольку $B_0(2, 5, 7)$ имеет достаточно большой порядок (он равен 5^{18}), для нахождения диаметра данной группы необходимо применять суперкомпьютерные вычисления. В настоящее время идет работа по

программной реализации параллельной версии алгоритма на языке C++ с использованием интерфейса передачи данных MPI. Вычисления будут проводиться на суперкомпьютере Сибирского федерального университета.

ЛИТЕРАТУРА

1. *Halt D., Eick B., and O'Brien E.* Handbook of computational group theory. Boca Raton: Chapman & Hall/CRC Press, 2005.
2. *Кузнецов А. А., Шлёпкин А. К.* Сравнительный анализ бернсайдовых групп $B(2, 5)$ и $B_0(2, 5)$ // Тр. Ин-та математики и механики УрО РАН. 2009. № 2. С. 125–132.

УДК 519.7

КОНВЕЙЕРИЗАЦИЯ КОМБИНАЦИОННЫХ СХЕМ

Ю. В. Поттосин, С. Н. Кардаш

Рассматривается вопрос повышения быстродействия устройства без памяти, преобразующего последовательность дискретных сигналов. Поставлена задача разбиения заданной многоуровневой комбинационной схемы на заданное число каскадов, на выходах которых должны быть поставлены регистры для обеспечения конвейерной обработки поступающих сигналов. Для решения этой задачи используется модель, основанная на представлении комбинационной схемы в виде ориентированного графа.

Ключевые слова: комбинационная схема, конвейеризация, ориентированный граф.

Повышению производительности систем обработки информации всегда уделялось большое внимание. Одним из способов повышения производительности является применение структуры конвейерного типа, который имеет ещё название «трубопровод» (перевод с англ. слова «pipeline») [1]. Подобную структуру образуют несколько независимых процессоров, соединённых между собой последовательно.

При построении систем цифровой обработки сигналов в режиме реального времени широкое распространение получил систолический принцип организации вычислений [2]. Информация в систолическом процессоре распространяется по конвейеру подобно тому, как пульсирует кровь при сокращении систолы сердца. В данной работе излагается попытка найти способ повышения быстродействия путем конвейеризации многоуровневой комбинационной схемы, построенной на основе СБИС.

В многоуровневой схеме устройства задержка складывается из задержек элементов самой длинной цепочки. Пусть на вход комбинационной схемы поступает последовательность p наборов двоичных сигналов. Если T — время задержки схемы, то период смены сигналов не может быть меньше T . Время реакции устройства на данную последовательность в этом случае не меньше pT . Разобьём схему на k каскадов (C_1, C_2, \dots, C_k) , и если τ_C — время задержки самого медленно действующего каскада, то $T \leq k\tau_C$. На выходы каждого каскада поставим элементы задержки (триггеры D), пропускающие сигналы с выходов каскада по сигналу синхронизации. Этот же сигнал синхронизации определяет период смены сигналов на входе устройства, который должен быть не меньше суммы двух задержек: задержки τ_C и задержки τ_D элемента D ($\tau_{\text{clock}} \geq \tau_C + \tau_D$). Теперь время реакции устройства на упомянутую последовательность равно $(k + p)\tau_{\text{clock}}$.

Нижняя граница длины p последовательности наборов двоичных сигналов на входе устройства, при которой имеет место ускорение обработки сигналов, определяется

неравенством $(k + p)\tau_{\text{clock}} < pT$. Учитывая нижнюю границу периода следования сигналов синхронизации τ_{clock} , получим

$$p > \frac{k\tau_{\text{clock}}}{T - \tau_{\text{clock}}} \geq \frac{k(\tau_C + \tau_D)}{T - \tau_C - \tau_D}.$$

Заданную схему требуется разбить на заданное число k каскадов, чтобы обеспечить по возможности максимальное быстродействие при описанном конвейерном режиме.

В качестве модели схемы используется бесконтурный орграф $G = (V, A)$. Его вершины из множества V представляют логические элементы и входные полюсы схемы, а дуги из множества A показывают направления сигналов от выходов одних элементов к входам других. Каждой вершине $v \in V$ приписан вес $\tau(v)$, представляющий задержку соответствующего элемента. Вершины, соответствующие входам схемы, имеют вес 0.

Сформируем последовательность слоев L_1, L_2, \dots, L_m , представляющую собой упорядоченное разбиение множества вершин V орграфа G с таким свойством, что если вершина v принадлежит полуокрестности исхода $N^+(u)$ вершины u , то эти вершины находятся в разных слоях и слой, содержащий вершину u , предшествует в этой последовательности слою с вершиной v (не обязательно непосредственно). Если длины путей от входов схемы к её выходам различны, то такое разбиение не является единственным. Следует выбрать такой вариант разбиения на слои, чтобы сумма весов всех слоев была по возможности минимальной. Под весом слоя понимаем максимум весов вершин, принадлежащих данному слою.

Можно выделить два типа вершин орграфа G . К одному типу отнесем вершины, которые лежат на самых длинных путях в орграфе G . Они строго распределяются по слоям и не могут менять своё положение. Их назовём *неподвижными*. Положение в слоях других вершин, которые назовём *подвижными*, можно менять в определённых пределах, скажем, от слоя L_l до слоя L_r ($l < r$). Эти пределы устанавливаются с помощью алгоритма, подобного алгоритму топологической сортировки [3].

Для окончательного распределения вершин по слоям так, чтобы сумма весов слоёв была по возможности минимальной, предлагается следующий способ. Удалив из орграфа G неподвижные вершины вместе с инцидентными им рёбрами, получим орграф H , в каждой компоненте которого выделим вершину с максимальным весом. Эту вершину поместим в один из допустимых для неё слоёв с максимальным весом. Границы положения вершин при этом изменятся и некоторые вершины из подвижных перейдут в неподвижные. Дальнейшее распределение по слоям можно вести для каждой компоненты орграфа H описанным способом.

Все пути в орграфе приводятся к единой длине путем добавления новых вершин с нулевым весом. Каждому из слоёв соответствует множество значений веса, приписанных вершинам, принадлежащим данному слою. Максимальное значение веса в слое представляет собой задержку прохождения сигнала в этом слое. Заданную комбинационную схему надо разбить на заданное число каскадов с минимизацией задержки в самом медленно действующем каскаде. Каждый каскад представляет собой упорядоченное множество слоёв.

Рассматриваемая задача теперь сводится к следующему. Дана последовательность положительных чисел (a_1, a_2, \dots, a_n) . Требуется её разбить на заданное число k отрезков B_1, B_2, \dots, B_k , где $B_i = (a_{n_{i-1}+1}, \dots, a_{n_i})$, $i = 1, 2, \dots, k$, $n_0 = 0$, $n_k = n$. При этом надо, чтобы величина $\max\{S_1, S_2, \dots, S_k\}$, где $S_i = a_{n_{i-1}+1} + \dots + a_{n_i}$, была по

возможности минимальной. Элементы B_i соответствуют каскадам в заданной схеме. Предлагается алгоритм получения решения данной задачи, близкого к оптимальному.

ЛИТЕРАТУРА

1. Каган Б. М., Каневский М. М. Цифровые вычислительные машины и системы. М.: Энергия, 1973.
2. Кухарев Г. А., Шмерко В. П., Зайцева Е. Н. Алгоритмы и систолические процессоры для обработки многозначных данных. Минск: Наука і тэхніка, 1990.
3. Кнут Д. Искусство программирования для ЭВМ. Т. 1. Основные алгоритмы. М.: Мир, 1976.

УДК 519.7

АЛГОРИТМ ПОИСКА ЗАПРЕТОВ БУЛЕВЫХ ФУНКЦИЙ

Д. В. Рябоконт

Предложен алгоритм поиска запрета булевой функции, основанный на методе ветвей и границ и позволяющий находить некоторый запрет булевой функции, запрет минимальной длины или все запреты до заданной длины.

Ключевые слова: запрет булевой функции, граф де Брёйна.

Будем обозначать $P_2(n)$ множество всех булевых функций от n переменных.

Определение 1. Говорят, что булева функция $f \in P_2(n)$ имеет запрет $y_1 \dots y_m \in \{0, 1\}^m$, если система булевых уравнений

$$f(x_i, \dots, x_{n+i-1}) = y_i, \quad i = 1, \dots, m, \quad (1)$$

несовместна. Если для любых $m \in \mathbb{N}$ и $y_1 \dots y_m \in \{0, 1\}^m$ система (1) совместна, то функция f называется *функцией без запрета*.

Определение 2. Графом де Брёйна $G_n(f)$ функции $f \in P_2(n)$ называют граф, у которого вершины — это все булевы векторы длины $n - 1$, а дуги поставлены во взаимно однозначное соответствие всевозможным булевым векторам длины n так, что вектору $b_1 b_2 \dots b_n$ соответствует дуга, направленная от вершины $b_1 b_2 \dots b_{n-1}$ к вершине $b_2 b_3 \dots b_n$ и помеченная значением $f(b_1 b_2 \dots b_n)$.

Вершину графа $G_n(f)$ назовём *особой вершиной типа* $\delta \in \{0, 1\}$, если обе исходящие из неё дуги помечены значением δ . Будем говорить, что *вектор* $y_1 y_2 \dots y_n$ *является продолжением вектора* $x_1 x_2 \dots x_n$, если $y_1 \dots y_{n-1} = x_2 \dots x_n$.

Утверждение 1. В графе де Брёйна $G_n(f)$ функции $f \in P_2(n)$, $n > 1$, имеющей запрет, есть хотя бы одна особая вершина.

Доказательство. Среди всех запретов функции f выберем любой запрет минимальной длины m . Пусть это есть $e_1 \dots e_m$. Так как m — минимальная длина запрета, граф $G_n(f)$ является $(m - 1)$ -полным графом де Брёйна, т. е. для любой последовательности $y_1 \dots y_{m-1} \in \{0, 1\}^{m-1}$ существует путь, её несущий. Рассмотрим последовательность $e_1 \dots e_{m-1}$. Для неё в $G_n(f)$ есть несущий путь, пусть он оканчивается в вершине v . Так как $e_1 \dots e_m$ — запрет, из вершины v нет дуги, помеченной e_m , т. е. обе исходящие из v дуги имеют значение \bar{e}_m и, следовательно, v — особая вершина. ■

Задача 1. Предлагается следующий алгоритм поиска некоторого запрета функции $f \in P_2(n)$. Пусть M_f^0, M_f^1 — множества булевых векторов длины n , на которых функция f принимает значения 0 и 1 соответственно; через $S(V)$ для $V \subseteq \{0, 1\}^n$ будем обозначать множество всевозможных продолжений векторов из V .

Работу алгоритма можно представить как построение следующего двоичного дерева. Вершинам дерева соответствуют множества булевых векторов длины n , дугам — значения 0 и 1. Корню дерева сопоставим множество $\{0, 1\}^n$. На каждом последующем шаге выбирается вершина ветвления, которой приписано множество V наименьшей мощности, если таких вершин несколько, то выбираем любую из них. Из выбранной вершины порождаются два потомка: левому (правому) потомку приписывается множество $M_f^0 \cap S(V)$ (соответственно $M_f^1 \cap S(V)$) и ведущая к нему дуга помечается нулём (соответственно единицей).

Если некоторой вершине u соответствует пустое множество, то запрет найден — это последовательность меток дуг пути из корня в вершину u .

Если вершине u соответствует одноэлементное множество $\{x_1 \dots x_n\}$, то проверяем, является ли вектор $x_2 \dots x_n$ особой вершиной (типа δ) в графе де Брёйна функции f , и если да — запрет найден, это $e_1 \dots e_{m-1} \bar{\delta}$, где $e_1 \dots e_{m-1}$ — последовательность меток дуг пути от корня до вершины u . Эта проверка легко выполняется по вектору значений функции f : вычисляется $h = f(x_2, \dots, x_n, 0) \oplus f(x_2, \dots, x_n, 1)$ и, если $h = 0$, то $x_2 \dots x_n$ — особая вершина типа $f(x_2, \dots, x_n, 0)$.

После каждого ветвления проверяем, есть ли среди полученных вершин неперспективные. Вершина объявляется неперспективной (и ветвление из неё не производится), если её метка (сопоставленное ей множество векторов) совпадает с меткой какой-либо вершины, встречавшейся ранее. Стоит заметить: это не значит, что при ветвлении из неперспективной вершины не будет найден запрет; это будет повторение одних и тех же действий. Для быстрого поиска вершины с заданной меткой будем хранить пройденные вершины в списке, индексированном по мощностям меток.

Если на каком-то шаге все вершины дерева объявлены неперспективными, а запрет ещё не найден, то у функции нет запрета.

По сравнению с алгоритмом 1, приведённым в [1], этот алгоритм более пригоден для практической реализации, так как в нём все шаги детально проработаны.

Задача 2. Для поиска запрета минимальной длины в исходном алгоритме нужно при нахождении запрета не завершать работу, а запомнить запрет и продолжать построение дерева таким же образом, пока не найдётся другой запрет; если он короче имеющегося, то запоминаем его. Дерево строится до тех пор, пока не исключим все вершины. При этом меняется критерий неперспективности вершины: вершина v , метка которой совпадает с меткой встречавшейся ранее вершины u , считается неперспективной в случае, если длина пути от корня до v больше или равна длине пути от корня до u . Для этого в списке пройденных вершин наряду с меткой каждой вершины будем хранить длину пути от неё до корня. Если найден запрет длины l , то все вершины l -го яруса объявляются неперспективными.

Задача 3. Назовём запрет *тупиковым*, если никакое его подслово не является запретом. Задача поиска всех тупиковых запретов до заданной длины l решается так же, как задача 2, но с запоминанием всех найденных запретов. Вершина объявляется неперспективной, если её метка совпадает с меткой предка (в этом случае можем получить запрет, но он не будет тупиковым), или она находится на расстоянии l от корня.

Алгоритм поиска произвольного запрета был применён ко всем функциям от 2, 3 и 4 переменных; результаты приведены в табл. 1.

Т а б л и ц а 1
Количество функций, имеющих запрет

n	Кол-во функций с запретом	Кол-во функций без запрета
2	10	6
3	226	30
4	64954	582

В табл. 2 приведены данные о среднем времени работы алгоритма поиска произвольного запрета для функций с количеством аргументов больше 5.

Т а б л и ц а 2
Время работы алгоритма

n	Среднее время, с	n	Среднее время, с
5	0,00019	13	0,128
6	0,0004	14	0,307
7	0,00091	15	0,698
8	0,0021	16	1,548
9	0,0047	17	3,461
10	0,011	18	7,552
11	0,025	19	16,528
12	0,061	20	33,312
		21	77,999

В дальнейшем предполагается разработать и реализовать алгоритмы решения задач 1–3 с помощью обхода дерева в ширину и в глубину (поиск с возвратом) и сравнить трудоёмкости этих алгоритмов (по количеству построенных вершин дерева и сложности их обработки).

ЛИТЕРАТУРА

1. Колегов Д. Н. О булевых функциях без запрета // Вестник Томского государственного университета. Приложение. 2005. № 14. С. 58–60.

УДК 519.7

ОБ ЭФФЕКТИВНОМ ПРЕДСТАВЛЕНИИ ДИЗЬЮНКТИВНЫХ НОРМАЛЬНЫХ ФОРМ ДИАГРАММАМИ СПЕЦИАЛЬНОГО ВИДА¹

А. А. Семенов

Рассматриваются диаграммы нового типа, используемые для представления дизъюнктивных нормальных форм. Показано, что данный класс диаграмм можно применять для компактного представления баз конфликтных дизъюнктов, накапливаемых в процессе нехронологического DPLL-вывода.

Ключевые слова: ДНФ, решающие диаграммы, BDD, ZDD, дизъюнктивные диаграммы.

¹Работа выполнена при поддержке гранта РФФИ № 11-07-00377.

Рассматривается произвольная всюду определённая булева функция $g : \{0, 1\}^n \rightarrow \{0, 1\}$ и произвольная ДНФ $D(g)$, представляющая g :

$$D(g) = K_1 \vee \dots \vee K_m,$$

где K_j , $j \in \{1, \dots, m\}$, — конъюнкты над множеством булевых переменных $X = \{x_1, \dots, x_n\}$. Далее ДНФ $D(g)$ будем рассматривать также как множество конъюнктов $\mathcal{D}(g) = \{K_1, \dots, K_m\}$. Зафиксируем на X порядок τ ($x_1 \prec x_2 \prec \dots \prec x_n$). Каждый конъюнкт K_j , $j \in \{1, \dots, m\}$, рассматривая его как множество литералов, упорядочим относительно τ .

Выделим в множестве $\mathcal{D}(g)$ подмножество конъюнктов, первый литерал в которых принадлежит множеству $L_{x_1} = \{x_1, \bar{x}_1\}$. Обозначим полученное множество конъюнктов через \mathcal{D}_{x_1} . Рассмотрим множество $\mathcal{D}^1 = \mathcal{D}(g) \setminus \mathcal{D}_{x_1}$ (полагаем, что оно не пусто). Обозначим через X^1 множество булевых переменных, фигурирующих в конъюнктах из $\mathcal{D}(g) \setminus \mathcal{D}_{x_1}$. Очевидно, что $X^1 \subseteq X \setminus \{x_1\}$. Пусть x_{i_1} — первая в смысле порядка τ переменная в X^1 . Выделим в $\mathcal{D}(g) \setminus \mathcal{D}_{x_1}$ подмножество конъюнктов, первый литерал в которых принадлежит множеству $L_{x_{i_1}} = \{x_{i_1}, \bar{x}_{i_1}\}$. Обозначим полученное множество конъюнктов через $\mathcal{D}_{x_{i_1}}$. Рассматриваем множество $\mathcal{D}^2 = \mathcal{D}^1 \setminus \mathcal{D}_{x_{i_1}}$. Далее поступаем по аналогии. Процесс продолжается до тех пор, пока для некоторого $k \geq 1$ не выполнится $\mathcal{D}^k = \emptyset$, $\mathcal{D}^k = \mathcal{D}^{k-1} \setminus \mathcal{D}_{x_{i_{k-1}}}$ (считаем, что $\mathcal{D}^0 = \mathcal{D}(g)$, $x_{i_0} = x_1$). В итоге имеем представление $\mathcal{D}(g)$ в виде

$$\mathcal{D}(g) = \bigcup_{k \in \{1, \dots, t\}} \mathcal{D}_{x_k}, \quad (1)$$

$\{1, \dots, t\} \subseteq \{1, \dots, n\}$, которому соответствует следующее представление $D(g)$:

$$D(g) = \bigvee_{k \in \{1, \dots, t\}} D_{x_k}, \quad (2)$$

где D_{x_k} — ДНФ, составленная из конъюнктов, входящих в \mathcal{D}_{x_k} .

Не ограничивая общности, рассмотрим D_{x_1} . Пусть $\{x_1^1, x_2^1, \dots, x_{p_1}^1\}$ — множество переменных, фигурирующих в D_{x_1} , $x_1^1 = x_1$. Представим D_{x_1} в виде

$$D_{x_1} = x_1 \cdot \Psi_{x_1} \vee \bar{x}_1 \cdot \Psi_{\bar{x}_1}, \quad (3)$$

где Ψ_{x_1} и $\Psi_{\bar{x}_1}$ — некоторые ДНФ над множеством переменных $\{x_2^1, \dots, x_{p_1}^1\}$. Представим каждую из этих ДНФ в форме (2), используя порядок τ .

Сделаем аналогичные действия для всех остальных D_{x_k} , входящих в правую часть (2). Продолжим описанный процесс рекурсивно.

Отметим, что каждое \mathcal{D}_{x_k} , входящее в (1), можно представить в виде r -арного дерева T_{x_k} , $r \leq n-1$. Листья дерева T_{x_k} помечаются либо символом «1», либо символом «?». Корень дерева T_{x_k} помечается x_k . Внутренние вершины каждого пути из корня в лист помечаются переменными из множества X . Для обозначения вершин с приписанными им переменными будем использовать выражения вида $v(x_s)$. Каждый путь в T_{x_k} из корня в лист, помеченный «1», соответствует конъюнкту из \mathcal{D}_{x_k} . Рёбра, выходящие из вершин, могут быть двух типов — пунктирные и сплошные. Если путь, проходящий через вершину $v(x_s)$ из корня T_{x_k} в лист, помеченный «1», определяет конъюнкт, в который x_s входит в виде литерала \bar{x}_s , то соответствующее ребро из $v(x_s)$ является пунктирным. Если же данный путь определяет конъюнкт, в который x_s входит без

отрицания, то соответствующее ребро является сплошным. Выходная степень каждой вершины (за исключением листьев) не меньше 2, причём из каждой вершины выходят как пунктирные, так и сплошные ребра.

Из (1) и определения $T_{x_k}, k \in \{1, \dots, t\}$, следует, что ДНФ $D(g)$ представима в виде леса, состоящего из t деревьев. Данный лес называется дизъюнктивным лесом.

Пример 1. Построим дизъюнктивный лес по ДНФ

$$x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4 \vee x_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \vee \bar{x}_1 \cdot x_2 \cdot x_4 \vee \bar{x}_1 \cdot \bar{x}_4 \vee x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \vee \bar{x}_2 \cdot x_4 \vee x_3 \cdot \bar{x}_4.$$

Фиксируем порядок $x_1 \prec x_2 \prec x_3 \prec x_4$. Представление (2) имеет вид

$$\begin{aligned} D_{x_1} &= x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4 \vee x_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \vee \bar{x}_1 \cdot x_2 \cdot x_4 \vee \bar{x}_1 \cdot \bar{x}_4, \\ D_{x_2} &= x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \vee \bar{x}_2 \cdot x_4, \\ D_{x_3} &= x_3 \cdot \bar{x}_4. \end{aligned}$$

Представление (3), например, для ДНФ D_{x_1} выглядит следующим образом:

$$x_1 \cdot (x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \vee \bar{x}_2 \cdot \bar{x}_3 \cdot x_4) \vee \bar{x}_1 \cdot (x_2 \cdot x_4 \vee \bar{x}_4).$$

Итогом применения описанной рекурсивной процедуры к рассматриваемой ДНФ является дизъюнктивный лес (рис. 1).

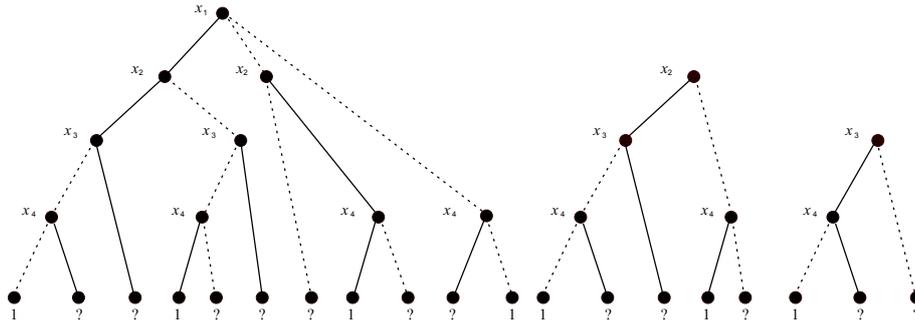


Рис. 1. Дизъюнктивный лес ДНФ из примера 1

Отметим, что идея представления ДНФ в форме дизъюнктивных лесов близка к идеям использования древовидных структур для представления булевых функций и множеств (соответственно BDD [1] и ZDD [2]). Однако имеются и существенные отличия. Во-первых, число путей в дизъюнктивном лесе заведомо ограничено полиномом от $n \cdot t$. Во-вторых, дизъюнктивный лес нельзя рассматривать как дерево решений (decision tree) булевой функции — пути, ведущие в ?-листья, вообще говоря, не дают наборы значений, на которых функция g принимает значение 0. В следующем частном случае, однако, дизъюнктивный лес совпадает с деревом решений.

Утверждение 1. Пусть $D(g)$ — СДНФ булевой функции g . Тогда дизъюнктивный лес, построенный по $D(g)$ в соответствии с описанными выше правилами, — это бинарное дерево, совпадающее с деревом решений g . В этом случае любой путь из корня в ?-лист определяет множество наборов значений переменных, на которых функция g принимает значение 0.

Используя структуры данных для представления КНФ/ДНФ, описанные в [3], можно показать, что справедлив следующий факт.

Теорема 1. Существует алгоритм, который по произвольной ДНФ $D(g)$ строит представляющий её дизъюнктивный лес за время $O(|D(g)|)$, где $|D(g)|$ — объём двоичного кода $D(g)$.

Пусть $D(g)$ — произвольная ДНФ и $F(D(g))$ — представляющий её дизъюнктивный лес. Каждую вершину в $F(D(g))$, за исключением терминальных вершин «1» и «?», можно задать следующим набором координат:

$$v(x) = \langle x, v_1^l, \dots, v_{k_l}^l, v_1^h, \dots, v_{k_h}^h \rangle.$$

Здесь $v_1^l, \dots, v_{k_l}^l$, $k_l \leq n - 1$, — вершины, которые являются детьми $v(x)$ по пунктирным рёбрам (по аналогии с используемой при описании BDD терминологией называем эти вершины low-детьми); $v_1^h, \dots, v_{k_h}^h$, $k_h \leq n - 1$, — вершины, которые являются детьми вершины $v(x)$ по сплошным рёбрам (high-дети).

По аналогии с процедурами построения BDD и ZDD можем определить операции склеивания вершин произвольного дизъюнктивного леса — склеиваются вершины, имеющие одинаковые наборы координат. Подобным же образом определяются операции сокращения. Результатом применения к дизъюнктивному лесу $F(D(g))$ склеивания и сокращения является дизъюнктивная диаграмма, обозначаемая через $\Delta(D(g))$.

Теорема 2. Существует алгоритм, который строит по произвольной ДНФ представляющую её дизъюнктивную диаграмму за время $O(n \cdot |D(g)|)$.

На рис. 2 приведена дизъюнктивная диаграмма, полученная в результате применения операций склеивания и сокращения к дизъюнктивному лесу рис. 1.

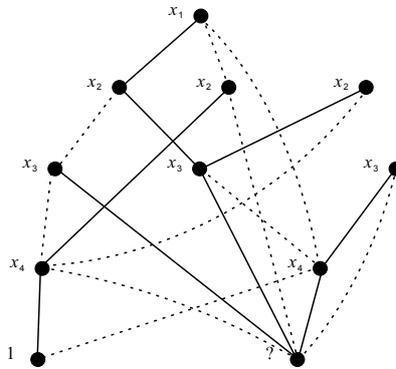


Рис. 2. Дизъюнктивная диаграмма для ДНФ из примера 1

Основное предназначение дизъюнктивных диаграмм — компактное представление конфликтных баз, накапливаемых в процессе нехронологического DPLL-вывода. Напомним, что в соответствии с [4] нехронологический DPLL сохраняет информацию о пройденных участках дерева поиска в виде булевых ограничений, называемых конфликтными дизъюнктами. Если C — исходная КНФ, к которой применяется DPLL, и D — конфликтный дизъюнкт, то C выполнима тогда и только тогда, когда выполнима КНФ $C \cdot D$. Современные SAT-решатели, использующие нехронологический DPLL, могут накапливать очень большие объёмы конфликтных дизъюнктов. Заполнение конфликтной информацией оперативной памяти негативно сказывается на эффективности вывода. Таким образом, актуальна разработка таких приёмов модификации конфликтных баз, которые позволяют существенно снижать объём оперативной памяти, занимаемой этими базами.

В [5, 6] описан гибридный SAT+ROBDD вывод, в котором для представления конфликтных баз используются ROBDD. В работе [6] определён специальный тип вывода, представляющий собой ROBDD-аналог итеративного применения правила Unit Propagation. Основным минус такого подхода — необходимость согласования двух структур данных, ROBDD и КНФ, что неизбежно сказывается на скорости работы программы. При этом несложно показать, что переход от ROBDD, представляющей конфликтную базу, к КНФ в соответствии со стандартными техниками (см., например, [7]) приводит к тому, что на конфликтной части КНФ в общем случае перестает работать FDA-процедура (Failure Driven Assertion [4]), которая является одним из базовых элементов во всех современных нехронологических DPLL-решателях.

Покажем, что данную проблему можно эффективно решить, если вместо ROBDD использовать дизъюнктивные диаграммы.

Теорема 3. Пусть C' — произвольная КНФ над множеством переменных X , D — произвольный дизъюнкт, входящий в C' , и $L_D = \{x_{i_1}^{\alpha_{i_1}}, \dots, x_{i_s}^{\alpha_{i_s}}\}$ — множество литералов данного дизъюнкта. Пусть $\Delta(\overline{C'})$ — дизъюнктивная диаграмма, представляющая ДНФ $\overline{C'}$, число вершин в которой есть N . Существует алгоритм, который за время $O(N)$ строит по $\Delta(\overline{C'})$ КНФ \tilde{C} над множеством переменных Y , $X \subset Y$, такую, что подстановка в \tilde{C} произвольного множества литералов вида $\{x_{j_1}^{\alpha_{j_1}}, \dots, x_{j_{s-1}}^{\alpha_{j_{s-1}}}\}$, $\{j_1, \dots, j_{s-1}\} \subset \{i_1, \dots, i_s\}$, даёт вывод по правилу Unit Propagation литерала из $L_D \setminus \{x_{j_1}^{\alpha_{j_1}}, \dots, x_{j_{s-1}}^{\alpha_{j_{s-1}}}\}$.

Предположим, что C' — конфликтная база, накопленная нехронологическим DPLL-выводом при решении проблемы выполнимости КНФ C . Пусть D — произвольный конфликтный дизъюнкт. Теорема 3 утверждает наличие эффективного алгоритма, строящего по дизъюнктивной диаграмме $\Delta(\overline{C'})$ КНФ \tilde{C} , которая может сама по себе использоваться в роли базы конфликтных ограничений. При этом на \tilde{C} работает FDA-процедура, то есть подстановка в \tilde{C} отрицаний произвольных $s - 1$ литералов дизъюнкта D даёт вывод по правилу Unit Propagation оставшегося литерала из D . Особо отметим, что на практике объём памяти, занимаемый КНФ \tilde{C} , может быть существенно (в некоторых ситуациях в разы) меньше, чем объём, занимаемый C' .

ЛИТЕРАТУРА

1. Bryant R. E. Graph-based algorithms for Boolean function manipulation // IEEE Trans. Comput. 1986. V. 35. No. 8. P. 677–691.
2. Minato S. Zero-suppressed BDDs for set manipulation in combinatorial problems // Proc. DAC-93, June 14–18, 1993, Dallas, Texas. P. 272–277.
3. Dowling W. and Gallier J. Linear-time algorithms for testing the satisfiability of propositional Horn formulae // J. Logic Programming. 1984. V. 1. No. 3. P. 267–284.
4. Marques-Silva J. P. and Sakallah K. A. GRASP: A search algorithm for propositional satisfiability // IEEE Trans. Comput. 1999. V. 48. No. 5. P. 506–521.
5. Игнатьев А. С., Семенов А. А. Алгоритмы работы с ROBDD как с базами булевых ограничений // Прикладная дискретная математика. 2010. № 1. С. 86–104.
6. Ignatiev A. and Semenov A. DPLL+ROBDD derivation applied to inversion of some cryptographic functions // LNCS. 2011. V. 6695. P. 76–89.
7. Een N. and Sorensson N. Translating pseudo-Boolean constraints into SAT // J. Satisfiability, Boolean Modeling and Computation. 2006. V. 2. P. 1–25.

УДК 511.9

РЕАЛИЗАЦИЯ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА ПОИСКА КРАТЧАЙШЕГО ВЕКТОРА В БЛОЧНОМ МЕТОДЕ КОРКИНА — ЗОЛОТАРЕВА

В. С. Усатюк

Предложена параллельная реализация алгоритма Каннана для решения задач поиска кратчайшего и короткого векторов в решётке. Алгоритм может применяться как в составе блочного метода Коркина — Золотарева, так и независимо. Эксперимент показал трёхкратное ускорение работы блочного метода Коркина — Золотарева на четырёхъядерной системе.

Ключевые слова: решётка, проблема поиска кратчайшего вектора, блочный метод Коркина — Золотарева.

Определение 1. Базис $B = \{b_1, b_2, \dots, b_m\}$ решётки $L \subset R^n$ приведён блочным методом Коркина — Золотарева (BKZ, Block Korkin — Zolotarev method [1]) с блоком β , если:

- 1) базис B приведён по длине;
- 2) $\|b_i^\perp\| = \lambda_1(L_i)$, $i = 1, \dots, m$, где $\lambda_1(L_i)$ — длина кратчайшего вектора в обратной (сопряжённой) решётке L_i , образованной ортогональным дополнением векторного пространства с базисом $b_i, \dots, b_{\min(i+\beta-1, m)}$.

BKZ-метод содержит две основные процедуры, которые могут быть распараллелены: ортогонализация базиса решётки и поиск кратчайшего вектора. Вопрос распараллеливания алгоритмов ортогонализации базиса рассматривался в работе [2].

Алгоритм поиска кратчайшего вектора представляет собой вариант метода ветвей и границ и заключается в переборе линейных комбинаций базисных векторов решётки, дающих вектор с нормой, ограниченной сверху некоторым положительным числом A , которое может уменьшаться в процессе поиска.

На начальном этапе можно принять $A = \min \left\{ \sqrt{\gamma_m} \cdot (\det L)^{\frac{1}{m}}, \|b_*\| \right\}$, где γ_m — константа Эрмита; b_* — наименьший по норме вектор в исходном базисе решётки.

Можно показать [3, 4], что координаты кратчайшего вектора удовлетворяют следующей системе неравенств:

$$\begin{cases} x_m^2 \|b_m^\perp\|^2 \leq A^2, \\ (x_{m-1} + \mu_{m,m-1}x_m) \|b_{m-1}^\perp\|^2 \leq A^2 - x_m^2 \|b_m^\perp\|^2, \\ \dots \\ (x_1 + \sum_{i=2}^m x_i \mu_{i,j})^2 \|b_1^\perp\|^2 \leq A^2 - \sum_{j=2}^m l_j, \end{cases} \quad (1)$$

где $\{b_j^\perp\}_{j=1}^m$ — ортогональный базис решётки, полученный из исходного базиса, например в результате процедуры ортогонализации Грама — Шмидта (через μ_{ij} обозначены коэффициенты Грама — Шмидта), $l_j = \left(x_j + \sum_{i=j+1}^m x_i \mu_{i,j} \right)^2 \|b_j^\perp\|^2$.

Для решения системы (1) используем метод ветвей и границ, то есть организуем поиск решения в виде дерева. Корень дерева помечен переменной x_m . Из корня выходит N ветвей, каждая из которых соответствует конкретному решению первого неравенства в (1). Пусть α_m — произвольное решение этого неравенства. Ветвь дерева

поиска, соответствующую данному решению, будем называть α_m -ветвью. Следующая вершина α_m -ветви помечается переменной x_{m-1} . Данная вершина является корнем дерева поиска для системы неравенств, полученной из (1) подстановкой $x_m = \alpha_m$. Если хотя бы одно из неравенств в (1) при этом не выполняется, то рассматриваемая ветвь отсекается (поскольку заведомо не содержит решения, дающего меньшую верхнюю границу, чем текущая). Описанный процесс по аналогии продолжается далее. Если найден вектор со значением нормы, меньшим текущей верхней границы, то значение верхней границы (рекорд) обновляется.

Описанный алгоритм реализован с использованием потоковой модели NPTL (Native POSIX Thread Library [5]) под CentOS 6.3. Каждый из потоков осуществляет вычисление своей α_m -ветви, исходящей из корня дерева.

На задаче приведения 103-мерной решётки ВКЗ-методом с размером блока $\beta = 52$ на четырёх потоках, выполняемых на AMD Phenom 965/8 Gb DDR2-800, он продемонстрировал трёхкратное ускорение в сравнении с последовательным ВКЗ-решателем `fp111-4.0.1` [6]. Предложенный алгоритм применялся в составе решателя, который занял 1-е и 6-е места на международных конкурсах поиска коротких векторов [7] для норм $A = m \cdot \det(L)^{1/m}$ и $A = 1,05 \frac{\Gamma(m/2 + 1)^{1/m}}{\sqrt{\pi}} \det(L)^{1/m}$ соответственно.

ЛИТЕРАТУРА

1. *Schnorr C. P.* Block reduced lattice bases and successive minima // *Combinatorics, Probability and Computing*. 1994. V. 3. P. 507–522.
2. *Усатюк В. С.* Реализация параллельных алгоритмов ортогонализации в задаче поиска кратчайшего базиса целочисленной решетки // *Прикладная дискретная математика. Приложение*. 2012. № 5. С. 120–122.
3. *Kannan R.* Improved algorithms for integer programming and related lattice problems // *Proc. STOC'83*. New York, NY, USA, 1983. P. 193–206.
4. *Hanrot G. and Stehle D.* Improved analysis of Kannan's shortest lattice vector algorithm // *LNCS*. 2007. V. 4622. P. 170–186.
5. *Kerrisk M.* The Linux programming interface: a Linux and UNIX system programming handbook. San Francisco, USA: No Starch Press, 2010. 1552 p.
6. <http://perso.ens-lyon.fr/damien.stehle/fp111/index.html> — Приложение `fp111`. 2013.
7. <http://www.latticechallenge.org/ideallattice-challenge/index.php> — Ideal lattice challenge (SVP, Approx-SVP). 2012.

УДК 004.627

РАСПАРАЛЛЕЛИВАНИЕ АЛГОРИТМА ДЕКОДИРОВАНИЯ СТАНДАРТА СЖАТИЯ ВИДЕОДАННЫХ H.265/HEVC

Р. И. Черняк

Рассматривается возможность параллельной реализации декодера в новом стандарте сжатия цифровых видеопоследовательностей H.265/HEVC. Предложены два способа параллелизации декодера, эффективность которых обоснована теоретически и показана экспериментально.

Ключевые слова: *видеокодирование, сжатие цифрового видео, распараллеливание алгоритмов декодирования.*

Новейший стандарт сжатия видео H.265/HEVC [1], предложенный в 2013 г. совместно группой экспертов по видеокодированию VCEG (Video Coding Experts Group) и экспертной группой по движущемуся изображению MPEG (Moving Picture Experts Group), представляет собой очередной шаг в технологиях видеокомпрессии. Он призван заменить широко используемый в настоящее время стандарт H.264/AVC [2]. Ключевым требованием, предъявляемым к HEVC, является двукратное уменьшение объёма сжатого видео (при фиксированном качестве) по сравнению с AVC. Для достижения данной цели разработчиками стандарта предложены новые и доработаны существующие подходы к сжатию видео. В результате цель достигнута, однако, в связи с усложнением логики работы кодека, значительно увеличилась его алгоритмическая сложность. Поскольку в приложениях критически важной является задача обеспечения кодирования и декодирования видеопотока в реальном времени (около 25 кадров в секунду), возникает задача эффективной реализации стандарта HEVC. Одним из наиболее действенных на сегодня способов увеличения скорости работы кодека на современных устройствах является параллельная реализация некоторых его алгоритмов. В данной работе речь пойдёт о параллельных алгоритмах декодирования в рамках стандарта H.265/HEVC.

1. Краткое описание кодека H.265/HEVC

Как и в предыдущих стандартах видеокомпрессии, в HEVC кодирование видеопоследовательности осуществляется по кадрам. Базовой единицей кодирования является CU (*coding unit*) — квадратный блок 16×16 , 32×32 или 64×64 пикселей. В первом приближении алгоритм кодирования можно описать следующим образом. Входящий кадр разбивается на CU, затем последовательно кодируется каждый из этих блоков. Более формально, кодек HEVC реализует классическую схему «гибридного» кодирования [3]. Не вдаваясь в её детали, отметим лишь, что завершающей стадией кодирования каждого CU является его энтропийное сжатие арифметическим кодером SBAC (*Syntax-based context-adaptive Binary Arithmetic Coding*). Соответственно первым шагом декодирования является выделение из битового потока данных, соответствующих очередному CU, их энтропийное декодирование, а затем окончательное восстановление (*реконструирование*). SBAC представляет собой контекстно-адаптивный энтропийный кодер, это значит, что при кодировании (декодировании) очередного CU используется информация о состоянии кодера в определённый момент (*контекст*). При обработке каждого CU видеопоследовательности контекст изменяется. Таким образом энтропийное кодирование и декодирование кадра осуществляется последовательно по всем CU.

2. Способы распараллеливания

В рамках данной работы предложены два механизма распараллеливания декодера HEVC. Первый представляет собой универсальный способ, который подходит для любых видеоданных, закодированных HEVC. Второй использует новую особенность стандарта H.265, в связи с этим он может быть использован только на входных видеоданных, закодированных специальным образом.

2.1. Распараллеливание по строкам

При декодировании типичного кадра Full HD-видео (кадр размера 1920×1080 пикселей) энтропийное декодирование занимает около 10 % времени работы кодека. Реконструирование изображения при этом занимает около 65 %; оставшиеся 25 % — вспомогательные операции инициализации данных. Энтропийное декодирование осуществ-

ляется сугубо последовательно; реконструирование может выполняться параллельно. Единственным требованием в этом случае является следующее.

Требование 1. Для корректной работы декодера при реконструировании CU на позиции (i, j) необходимо, чтобы уже были реконструированы четыре его «соседа»: $(i - 1, j)$; $(i - 1, j - 1)$; $(i, j - 1)$ и $(i - 1, j + 1)$.

Рис. 1 иллюстрирует данное условие. Темным цветом выделены уже реконструированные CU, светло-серым — находящиеся в процессе реконструкции, белым — ожидающие своей очереди. В данном примере реконструирование осуществляется тремя вычислителями параллельно.

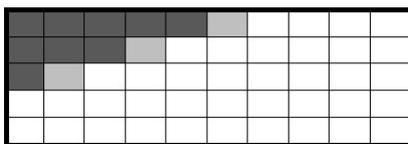


Рис. 1. Параллельное реконструирование CU в кадре

Предлагаемый механизм распараллеливания HEVC-декодера основывается на следующих двух идеях:

- 1) отделение процесса энтропийного декодирования от реконструирования;
- 2) распараллеливание реконструирования с учетом требования 1.

Предлагаемый алгоритм можно описать следующим образом. Пусть имеется один основной поток выполнения и пул из $(n - 1)$ рабочих потоков. Пусть основной поток последовательно осуществляет энтропийное декодирование кадра; как только очередная строка декодирована энтропийным декодером, пулу потоков передается очередная задача на реконструирование этой строки. В свою очередь, каждый рабочий поток из пула производит проверку условия из требования 1 и, если оно выполнено, реконструирует очередной CU. Когда энтропийное декодирование кадра завершено, основной поток дожидается завершения работы всех потоков пула. Поскольку энтропийное декодирование осуществляется значительно быстрее, чем реконструирование, достаточно быстро будут задействованы все доступные рабочие потоки.

2.2. Распараллеливание по тайлам

Одним из новшеств стандарта H.265/HEVC стало обеспечение возможности параллельного энтропийного декодирования кадра. В общем случае SBAC осуществляет последовательное декодирование, однако разработчиками стандарта предложены способы распараллеливания этой процедуры. Разбитый на CU кадр представляется в виде матрицы $n \times t$, каждым элементом которой является множество CU. Такое разбиение называется *тайловым*, а элемент матрицы — *тайлом*. В начале каждого тайла в качестве контекста для энтропийного кодека используется значение по умолчанию. Таким образом, можно осуществлять процедуры кодирования и декодирования параллельно. Рис. 2 иллюстрирует возможное тайловое разбиение кадра и способ его декодирования.

В данном примере кадр разбивается на тайлы матрицей 2×2 . Дальнейшее декодирование осуществляется параллельно в каждом тайле. Такой способ позволяет распараллелить как энтропийное декодирование, так и реконструирование кадра. Таким образом, распараллеливание по тайлам предпочтительнее, чем по строкам, однако наличие тайлов — это специфичная настройка кодека, которую нельзя гарантировать, что мешает использовать данный подход в общем случае.

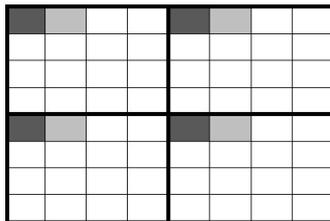


Рис. 2. Параллельное декодирование кадра по тайлам

Следует отметить, что поскольку на границе тайлов происходит сброс контекста энтропийного кодека, использование большого количества тайлов влечёт некоторое снижение качества его работы.

3. Эксперименты

В экспериментах в качестве входных данных использовались три различных видео-последовательности: анимированные фильмы Big Buck Bunny [4] и Sintel [5], свободно распространяемые организацией Blender Foundation [6], и видеозапись соревнований по серфингу, проходивших в 2012 г. во Вьетнаме [7]. Каждый видеоролик представлен в нескольких разрешениях: Big Buck Bunny — 1920×1080 (1080p), 1280×720 (720p), 640×360 (360p); Sintel — 1920×816 (816p), 1280×408 (408p), 640×272 (272p); Serfing — 1280×720 (720p), 640×360 (360p). Длительности роликов составляют 14315, 8883 и 21313 кадров соответственно.

Эксперименты проводились для трёх алгоритмов декодирования: 1) последовательного; 2) параллельного по строкам; 3) параллельного по тайлам. В качестве аппаратной платформы использовался ПК с четырёхъядерным процессором Intel Core I7 3770 и 8GB ОЗУ. В табл. 1 приведены результаты экспериментов. Скорость декодирования представлена в кадрах в секунду.

Т а б л и ц а 1

Скорость декодирования различных входных данных последовательным и параллельными алгоритмами

Входные данные	Последовательное	Парал. по строкам	Парал. по тайлам
Big Buck Bunny 1080p	20,4	38,7	43,1
Big Buck Bunny 720p	44,6	79,1	89,5
Big Buck Bunny 360p	159	201,6	242,6
Sintel 816p	23,4	47,5	52,6
Sintel 408p	83,3	128,4	154,4
Sintel 272p	172,3	217,5	263,1
Serfing 720p	23,8	50,5	56,2
Serfing 360p	80,0	123,4	155,8

Согласно результатам экспериментов, наибольшее ускорение наблюдается при применении тайловой модели распараллеливания на видеопоследовательностях высоких разрешений. С убыванием размера кадра ускорение уменьшается.

Вычислим теоретическое ускорение для тайловой и строковой моделей распараллеливания при декодировании Full HD-видео. Энтропийное декодирование, реконструирование и вспомогательные операции в этом случае занимают 10, 65 и 25 % времени декодирования соответственно. Пусть t — общее время последовательного декодирования, тогда $0,65t$ — время последовательного реконструирования. При строковом

распараллеливании на четыре ядра время реконструирования уменьшается в 4 раза. Таким образом, общее время декодирования может быть вычислено по формуле $0,65t/4 + 0,1t + 0,25t = 0,5125t$. Поскольку скорость декодирования обратно пропорциональна времени, максимальный коэффициент ускорения декодирования составляет 1,95. Вычисленный аналогичным образом коэффициент ускорения для тайлового распараллеливания составляет 2,29. Заметим, что экспериментально полученные значения ускорений несколько меньше теоретических. Это объясняется тем, что при обеих моделях распараллеливания данные зависят друг от друга. В случае строковой модели эта зависимость определяется требованием 1, в случае тайловой — потенциально различной сложностью каждого тайла.

Как отмечалось ранее, с ростом количества тайлов ухудшается эффективность работы энтропийного кодера. Проведён эксперимент с целью определения степени ухудшения. Одно и то же видео кодировалось в несколько тайлов при прочих одинаковых настройках кодирования. Табл. 2 иллюстрирует изменение качества видео, выраженного в виде пикового отношения сигнала к шуму (*PSNR*), и эффективности его сжатия (*bitrate*) в зависимости от количества тайлов. Видно, что ухудшение качества видео и степени сжатия пренебрежимо мало даже при кодировании каждого кадра в восемь тайлов. Эксперимент проводился для видео с разрешением 720р.

Таблица 2

Зависимость качества видео и степени сжатия от количества тайлов

Количество тайлов	1	2	3	4	5	6	7	8
Bitrate (kbps)	1456,2	1466,2	1474,4	1486,2	1494,9	1501,7	1507,7	1519,6
PSNR (dB)	41,448	41,444	41,442	41,440	41,437	41,434	41,433	41,432

Таким образом, теоретически и экспериментально показано, что использование модели тайлового распараллеливания даёт выигрыш по сравнению со строковым. Вместе с тем тайловая модель не может быть использована для произвольных данных. Представляется целесообразным использовать тайловое распараллеливание в случае, если входное видео закодировано в несколько тайлов, в противном случае стоит использовать строковый алгоритм. В случае тайлового распараллеливания строковый метод может быть использован внутри каждого тайла. Такой подход представляет интерес при декодировании на многоядерных устройствах с целью обеспечения высокой скорости декодирования путём максимальной загрузки каждого ядра.

ЛИТЕРАТУРА

1. *Bross B., Han W.-J., Ohm J.-R., et al.* High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Last Call). 12th Meeting: Geneva, CH, 14–23 Jan. 2013
2. ISO/IEC 14496-10:2003. Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding.
3. *Sullivan G. J., Ohm J.-R., Han W.-J., and Wiegand T.* Overview of the High Efficiency Video Coding (HEVC) standard // IEEE Trans. Circuits and Systems for Video Technology. 2012. V. 22. No. 12. P. 1649–1668.
4. <http://www.bigbuckbunny.org/> — Big Buck Bunny. 2013.
5. <http://www.sintel.org/> — Sintel, the Durian Open Movie Project. 2013.
6. <http://www.vungtau-city.com/?p=1558> — Russia Surfing Cup and International Kitesurfing Competitions in Vung Tau | Vung Tau City Portal. 2013.
7. <http://www.blender.org/> — blender.org — Home of the Blender project — Free and Open 3D creation software. 2013.

УДК 519.863

ТОЧНЫЙ АЛГОРИТМ ДЛЯ РЕШЕНИЯ ОДНОГО ЧАСТНОГО СЛУЧАЯ ЗАДАЧИ ВЕБЕРА В ДИСКРЕТНОЙ ПОСТАНОВКЕ

Р. Э. Шангин

Предлагается детерминированный квазиполиномиальный алгоритм, находящий точное решение задачи Вебера в дискретной постановке для n -последовательно-связной цепи и конечного множества позиций размещения, основанный на динамическом программировании. Дан теоретический анализ предложенного алгоритма. Проведен вычислительный эксперимент по анализу эффективности предложенного алгоритма в сравнении с пакетом IBM ILOG CPLEX.

Ключевые слова: задача Вебера, n -последовательно-связная цепь, динамическое программирование, точный алгоритм, квазиполиномиальный алгоритм.

Рассматривается задача Вебера в дискретной постановке [1] для n -последовательно-связной цепи [2] и конечного множества позиций размещения.

Пусть $G = (J, E)$ — n -последовательно-связная цепь, где J — множество вершин графа (размещаемые объекты), $E = \{(i, j) : i, j \in J\}$ — множество рёбер (связи между размещаемыми объектами). Пусть V — конечное множество, элементами которого являются позиции, предназначенные для размещения вершин графа G . Размещением вершин графа G назовем однозначное отображение $\pi : J \rightarrow V$, то есть вершина $i \in J$ размещается в позицию $\vartheta_i \in V$, причём в одну позицию возможно размещение нескольких вершин графа.

Стоимость размещения вершины $i \in J$ в множестве позиций V задается функцией $p : J \times V \rightarrow \mathbb{R}^+$, где $p(i, \vartheta_i)$ — стоимость размещения вершины $i \in J$ в позиции $\vartheta_i \in V$. Стоимость размещения ребра $(i, j) \in E$ на V^2 определяется функцией $c : E \times V^2 \rightarrow \mathbb{R}^+$, где $c((i, j), \vartheta_i, \vartheta_j)$ — стоимость размещения ребра $(i, j) \in E$ на V^2 при размещении его концевых вершин $i, j \in J$ в позициях $\vartheta_i, \vartheta_j \in V$ соответственно.

Требуется разместить вершины графа G в позициях множества V таким образом, чтобы суммарная стоимость размещения вершин и рёбер графа G была минимальной. Формулировка исследуемой задачи в терминах отображений имеет вид

$$F(\pi) = \sum_{(i,j) \in E} c((i,j), \pi(i), \pi(j)) + \sum_{i \in J} p(i, \pi(i)) \rightarrow \min_{\pi} . \quad (1)$$

Задача Вебера в дискретной постановке в случае, когда структура связей между размещаемыми объектами задается графом произвольной структуры, является NP-трудной [3], однако известны её полиномиально разрешимые частные случаи [1, 4, 5].

Предлагается квазиполиномиальный алгоритм, находящий точное решение задачи Вебера для n -последовательно-связной цепи и конечного множества позиций размещения, основанный на динамическом программировании (ДП).

Идея алгоритма заключается в следующем. Обозначим тройкой (G, V, F) задачу Вебера (1), где $G = (J, E)$ — n -последовательно-связная цепь; V — конечное множество позиций размещения и F — функция стоимости размещения графа G . На множестве вершин цепи G задается нумерация согласно определению 1, предложенному в работе [2]. Процесс решения задачи (G, V, F) разбивается на $|J| - n$ шагов процесса ДП, где переход от одного шага динамического процесса к другому рассматривается как последовательный перебор вершин графа G в соответствии с их нумерацией. Обозначим $G_i = (J_i, E_i)$ подграф графа G , индуцированный вершинами, номера которых принадлежат множеству $\{1, 2, \dots, i + n\}$, где $1 \leq i \leq |J| - n$. На каждом шаге

$i = 1, 2, \dots, |J| - n$ процесса ДП алгоритм рекуррентно находит оптимальное решение подзадачи (G_i, V, F) , зная оптимальное решение подзадачи (G_{i-1}, V, F) , полученное им на предыдущем шаге. Так, на конечном шаге $|J| - n$ алгоритм находит оптимальное решение исходной задачи (G, V, F) , поскольку $G_{|J|-n} = G$, а подзадачи (G_i, V, F) на предыдущих шагах решены оптимально. Доказана

Теорема 1. Предложенный алгоритм находит точное решение задачи Вебера (1), где G – n -последовательностьсвязная цепь; V – конечное множество позиций размещения.

Вычислительная сложность алгоритма не превосходит $O(|V|^{n+1}(|J| - n))$ операций, пространственная сложность равна $O(|V|^{n+2})$. Проведён вычислительный эксперимент по анализу времени работы алгоритма. Для оценки эффективности алгоритма использовался программный пакет IBM ILOG CPLEX Optimization Studio 12.2 (решение модели целочисленного линейного программирования задачи Вебера (1) алгоритмом ветвей и границ с ограничением по времени работы). Тестовые данные генерировались случайным образом с равномерным распределением. Решались серии из 30 задач одинакового размера $(|J|, |V|)$ – от (5,5) до (100,100).

Вычислительный эксперимент показал, что применение данного алгоритма перспективно при сравнительно малых значениях n , причём чем меньше n и чем больше количество вершин размещаемого графа, тем предлагаемый алгоритм более эффективен по сравнению с точными алгоритмами, являющимися вариациями полного перебора с отсевом заведомо бесперспективных подмножеств допустимых решений. Чем ближе n к числу вершин размещаемого графа, тем предлагаемый алгоритм менее эффективен. Например, при решении задачи Вебера для n -последовательностьсвязной цепи $G = (J, E)$ с количеством вершин $|J| = n + 1$ время работы предложенного алгоритма оказалось сравнимо с временем работы алгоритма, использующего идею тривиального последовательного перебора допустимых решений.

ЛИТЕРАТУРА

1. *Panyukov A. V. and Pelzwerger B. V.* Polynomial algorithms to finite Veber problem for a tree network // J. Comput. Appl. Math. 1991. V. 35. P. 291–296.
2. *Шангин Р. Э.* О некоторых свойствах n -последовательностьсвязной цепи // Вестник ЮУрГУ. Сер. Вычислительная математика и информатика. 2013. Т. 2. № 1. С. 106–113.
3. *Панюков А. В.* Модели и методы решения задач построения и идентификации геометрического размещения: дис. ... докт. физ.-мат. наук. М., 1999.
4. *Забудский Г. Г., Филимонов Д. В.* Решение дискретной минимаксной задачи размещения на сети // Изв. вузов. Математика. 2004. № 5. С. 33–36.
5. *Трубин В. А.* Эффективный алгоритм для задачи Вебера с прямоугольной метрикой // Кибернетика. 1978. № 6. С. 67–70.

СВЕДЕНИЯ ОБ АВТОРАХ

АБОРНЕВ Александр Викторович — ООО «Центр сертификационных исследований», г. Москва.

E-mail: aabornev@inbox.ru

АБРОСИМОВ Михаил Борисович — доцент, кандидат физико-математических наук, доцент Саратовского государственного университета, г. Саратов. E-mail: mic@rambler.ru

АГИБАЛОВ Геннадий Петрович — заведующий кафедрой защиты информации и криптографии Национального исследовательского Томского государственного университета, доктор технических наук, профессор. E-mail: agibalov@isc.tsu.ru

АЛЕХИНА Марина Анатольевна — профессор, доктор физико-математических наук, заведующая кафедрой Пензенского государственного университета, г. Пенза. E-mail: ama@sura.ru

АНАШКИНА Наталия Викторовна — кандидат технических наук, доцент, заместитель председателя УМС УМО по ИБ, г. Москва. E-mail: 6237030@mail.ru

АНИСЕНЯ Николай Ильич — студент кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.

E-mail: anisenya@gmail.com

АРБУЗОВ Дмитрий Сергеевич — студент кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.

E-mail: dsarbuzov@gmail.com

БАРСУКОВА Оксана Юрьевна — ассистент Пензенского государственного университета, г. Пенза. E-mail: dm@pnzgu.ru

БАТУЕВА Цындыма Чимит-Доржиевна — кандидат физико-математических наук, научный сотрудник Института математики им. С. Л. Соболева СО РАН, г. Новосибирск.

E-mail: batueva@math.nsc.ru

БОНДАРЕНКО Леонид Николаевич — доцент, кандидат технических наук, доцент кафедры дискретной математики Пензенского государственного университета, г. Пенза.

E-mail: leobond5@mail.ru

БОНДАРЕНКО Полина Павловна — студентка Саратовского государственного университета, г. Саратов. E-mail: polinabond@gmail.com

БОРОДИН Михаил Алексеевич — студент Московского государственного университета им. М. В. Ломоносова, г. Москва. E-mail: bor1m@mail.ru

БРОСЛАВСКИЙ Олег Викторович — студент кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.

E-mail: yalegko@isc.tsu.ru

БУЛАВИНЦЕВ Вадим Германович — программист Института динамики систем и теории управления СО РАН, г. Иркутск. E-mail: ichorid@mail.ru

БЫКОВА Валентина Владимировна — доцент, доктор физико-математических наук, профессор Института математики и фундаментальной информатики Сибирского федерального университета, г. Красноярск. E-mail: bykvalen@mail.ru

БЫЛКОВ Даниил Николаевич — кандидат физико-математических наук, ООО «Центр сертификационных исследований», г. Москва. E-mail: bilkov@gmail.com

ВАСИН Алексей Валерьевич — кандидат физико-математических наук, старший преподаватель кафедры дискретной математики Пензенского государственного университета, г. Пенза.

E-mail: alvarvasin@mail.ru

ВИТКУП Валерия Александровна — студентка механико-математического факультета Новосибирского государственного университета, г. Новосибирск. E-mail: vvitkup@yandex.ru

ВОЛГИН Артем Владимирович — преподаватель МГТУ МИРЭА, г. Москва.

E-mail: artem.volgin@bk.ru

ВОЛКОВ Денис Анатольевич — магистрант Омского государственного технического университета, г. Омск. E-mail: vol.denis54@gmail.com

ГЕУТ Кристина Леонидовна — ассистент Уральского государственного университета путей сообщения, г. Екатеринбург. E-mail: gluskokrl@rtural.ru

ДЕВЯНИН Петр Николаевич — доктор технических наук, доцент, председатель УМС УМО по Информационной безопасности, г. Москва. E-mail: peter_devyanin@hotmail.com

ЕВДОКИМОВ Александр Андреевич — профессор, заведующий лабораторией Института математики им. С. Л. Соболева СО РАН, г. Новосибирск. E-mail: evdok@math.nsc.ru

ЖАРКОВА Анастасия Владимировна — кандидат физико-математических наук, доцент кафедры теоретических основ компьютерной безопасности и криптографии Саратовского государственного университета, г. Саратов. E-mail: VAnastasiyaV@gmail.com

ЖУКОВ Кирилл Дмитриевич — сотрудник лаборатории ТВП, г. Москва.

ЗАЕЦ Мирослав Владимирович — сотрудник ФГУП «НИИ КВАНТ», г. Москва.

E-mail: mirzaets@hotmail.com

ЗАЙЦЕВ Георгий Юрьевич — студент кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.

E-mail: zaytsevgu@gmail.com

КАЛУЖИН Александр Константинович — студент Московского государственного университета им. М. В. Ломоносова, г. Москва. E-mail: alexskorp@yandex.ru

КАРДАШ Сергей Николаевич — кандидат технических наук, старший научный сотрудник Объединенного института проблем информатики НАН Беларуси, г. Минск.

КАРПУНИН Григорий Анатольевич — кандидат физико-математических наук, доцент факультета ВМК Московского государственного университета им. М. В. Ломоносова, г. Москва.

E-mail: karpunin@cs.msu.su

КАТЕРИНСКИЙ Денис Аркадьевич — студент кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.

E-mail: deniskat@isc.tsu.ru

КЛИМИНА Александра Сергеевна — студентка Сибирского государственного аэрокосмического университета им. акад. М. Ф. Решетнёва, г. Красноярск. E-mail: alkli@mail.ru

КОВАЛЕВ Дмитрий Сергеевич — аспирант кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, инженер-программист отдела технических средств и ремонта вычислительной техники ОАО «Информационные спутниковые системы» им. акад. М. Ф. Решетнёва, г. Железногорск.

E-mail: dmisk@hotmail.com, dmisk@iss-reshetnev.ru

КОЛЕГОВ Денис Николаевич — кандидат технических наук, доцент кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск. E-mail: d.n.kolegov@gmail.com

КОЛОМЕЕЦ Николай Александрович — аспирант Института математики им. С. Л. Соболева СО РАН, г. Новосибирск. E-mail: nkolomeec@gmail.com

КОМАРОВ Дмитрий Дмитриевич — аспирант, ассистент кафедры теоретических основ компьютерной безопасности и криптографии Саратовского государственного университета им. Н. Г. Чернышевского, г. Саратов. E-mail: KomarovDD@gmail.com

КОРЕНЕВА Алиса Михайловна — руководитель проектов по информационной безопасности ООО «Пойнтлэйн», г. Москва. E-mail: alisa.koreneva@gmail.com

КОРНИЕНКО Анастасия Сергеевна — ассистент Новосибирского государственного университета, г. Новосибирск. E-mail: anastasia.s.kornienko@gmail.com

КОЧЕМАЗОВ Степан Евгеньевич — программист Института динамики систем и теории управления СО РАН, г. Иркутск. E-mail: veinamond@gmail.com

КУЗНЕЦОВ Александр Алексеевич — профессор, доктор физико-математических наук, заведующий кафедрой Сибирского государственного аэрокосмического университета им. акад. М. Ф. Решетнёва, г. Красноярск. E-mail: alex_kuznetsov80@mail.ru

КУЗНЕЦОВА Александра Сергеевна — аспирантка Сибирского государственного аэрокосмического университета им. акад. М. Ф. Решетнёва, г. Красноярск. E-mail: alexakulch@rambler.ru

КУРГАНСКИЙ Алексей Николаевич — кандидат физико-математических наук, старший научный сотрудник Института прикладной математики и механики НАН Украины, г. Донецк. E-mail: topologia@mail.ru

КЯЖИН Сергей Николаевич — студент кафедры криптологии и дискретной математики Национального исследовательского ядерного университета МИФИ, г. Москва. E-mail: s.kyazhin@kaf42.ru

ЛИПСКИЙ Валерий Борисович — доцент кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, кандидат технических наук. E-mail: lipsky@mail.tsu.ru

ЛОСЕВ Александр Сергеевич — кандидат физико-математических наук, младший научный сотрудник Института прикладной математики ДВО РАН, г. Владивосток. E-mail: alexax@bk.ru

МЕДВЕДЕВ Никита Владимирович — ассистент Уральского государственного университета путей сообщения, г. Екатеринбург. E-mail: itcrypt@gmail.com

МЕДВЕДЕВА Наталья Валерьевна — кандидат физико-математических наук, доцент Уральского государственного университета путей сообщения, г. Екатеринбург. E-mail: medvedeva_n_v@mail.ru

МИРОНЕНКО Ольга Леонидовна — студентка кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск. E-mail: olga.l.kolcheva@gmail.com

МОДЕНОВА Ольга Владимировна — аспирантка кафедры теоретических основ компьютерной безопасности и криптографии Саратовского государственного университета, г. Саратов. E-mail: oginiel@rambler.ru

НАЖМИДЕНОВА Ажар Маратовна — студентка 1 курса магистратуры Новосибирского государственного университета, г. Новосибирск. E-mail: deviliona@yandex.ru

ОСИПОВ Дмитрий Юрьевич — студент Саратовского государственного университета им. Н. Г. Чернышевского, г. Саратов. E-mail: st_hill@mail.ru

ОСИПОВА Марина Анатольевна — доцент, кандидат физико-математических наук, научный сотрудник Института прикладной математики ДВО РАН, г. Владивосток. E-mail: mao1975@list.ru

ОТПУЩЕННИКОВ Илья Владимирович — кандидат технических наук, научный сотрудник Института динамики систем и теории управления СО РАН, г. Иркутск. E-mail: otilya@yandex.ru

ПАНКРАТОВА Ирина Анатольевна — доцент кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, кандидат физико-математических наук, доцент. E-mail: pank@isc.tsu.ru

ПЕСТУНОВ Андрей Игоревич — старший преподаватель Новосибирского государственного университета экономики и управления; научный сотрудник Института вычислительных технологий СО РАН, г. Новосибирск. E-mail: pestunov@gmail.com

ПОТАПКИН Алексей Игоревич — студент факультета информатики Национального исследовательского Томского государственного университета, г. Томск. E-mail: potapkin.alexey@gmail.com

ПОТТОСИН Юрий Васильевич — доцент, кандидат физико-математических наук, ведущий научный сотрудник Объединенного института проблем информатики НАН Беларуси, г. Минск. E-mail: pott@newman.bas-net.by

РЫБАКОВ Александр Сергеевич — кандидат физико-математических наук, сотрудник лаборатории ТВП, г. Москва.

РЯБОКОНЬ Денис Владимирович — студент кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.

E-mail: ryabokon.denis@ya.ru

САЛИЙ Вячеслав Николаевич — заведующий кафедрой теоретических основ компьютерной безопасности и криптографии Саратовского государственного университета им. Н. Г. Чернышевского, профессор, кандидат физико-математических наук, г. Саратов. E-mail: SaliiVN@info.sgu.ru

САФОНОВ Константин Владимирович — профессор, доктор физико-математических наук, заведующий кафедрой Сибирского государственного аэрокосмического университета им. акад. М. Ф. Решетнёва, г. Красноярск. E-mail: safonovkv@rambler.ru

СЕМЕНОВ Александр Анатольевич — кандидат технических наук, заведующий лабораторией Института динамики систем и теории управления СО РАН, г. Иркутск.

E-mail: biclop.rambler@yandex.ru

СТЕФАНЦОВ Дмитрий Александрович — старший преподаватель кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск. E-mail: d.a.stefantsov@isc.tsu.ru

ТИТОВ Сергей Сергеевич — доктор физико-математических наук, профессор Уральского государственного университета путей сообщения, г. Екатеринбург. E-mail: sergey.titov@usaaa.ru

ТКАЧЕНКО Николай Олегович — аспирант кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.

E-mail: n.o.tkachenko@gmail.com

ТОКАРЕВА Наталья Николаевна — кандидат физико-математических наук, старший научный сотрудник Института математики им. С. Л. Соболева СО РАН, г. Новосибирск.

E-mail: tokareva@math.nsc.ru

ТОРГАЕВА Татьяна Андреевна — студентка кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.

E-mail: t.a.torgaeva@gmail.com

ТУКТАРОВА Лидия Игоревна — студентка кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.

E-mail: ltuktarova@gmail.com

УСАТЮК Василий Станиславович — программист кафедры дискретной математики и защиты информации Братского государственного университета, г. Братск. E-mail: L@Lcrypto.com

ФИЛЮЗИН Станислав Юрьевич — студент механико-математического факультета Новосибирского государственного университета, г. Новосибирск. E-mail: forgogu@inbox.ru

ФОМИЧЕВ Владимир Михайлович — доктор физико-математических наук, профессор Финансового университета при Правительстве РФ, г. Москва. E-mail: fomichev@nm.ru

ФРОЛОВА Анастасия Александровна — магистрантка механико-математического факультета Новосибирского государственного университета, г. Новосибирск.

E-mail: frolova.anast@gmail.com

ЦИЦИАШВИЛИ Гурами Шалвович — профессор, доктор физико-математических наук, заместитель директора Института прикладной математики ДВО РАН, г. Владивосток.

E-mail: guram@iam.dvo.ru

ЧЕРЕМУШКИН Александр Васильевич — доктор физико-математических наук, член-корреспондент Академии криптографии РФ, заведующий кафедрой Института криптографии, связи и информатики, г. Москва. E-mail: avc238@mail.ru

ЧЕРНОВ Дмитрий Владимирович — аспирант кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.

E-mail: dm.vl.chernov@gmail.com

ЧЕРНЯК Роман Игоревич — аспирант кафедры защиты информации и криптографии Национального исследовательского Томского государственного университета, г. Томск.

E-mail: r.chernyack@gmail.com

ЧИЖОВ Иван Владимирович — кандидат физико-математических наук, ассистент кафедры математической кибернетики Московского государственного университета им. М. В. Ломоносова, г. Москва. E-mail: ivchizhov@gmail.com

ШАНГИН Роман Эдуардович — аспирант факультета вычислительной математики и информатики Южно-Уральского государственного университета, г. Челябинск.

E-mail: shanginre@gmail.com

ШОЛОМОВ Лев Абрамович — профессор, доктор физико-математических наук, главный научный сотрудник Института системного анализа РАН, г. Москва. E-mail: sholomov@isa.ru

ЩЕРБА Евгений Викторович — кандидат технических наук, доцент Омского государственного технического университета, г. Омск. E-mail: evscherba@gmail.com

АННОТАЦИИ ДОКЛАДОВ

SECTION 1

Abornev A. V. **DIGIT-INJECTIVE TRANSFORMATIONS OF A MODULE OVER A GALOIS RING.** New classes of nonlinear permutations that can be represented by linear transformations of a module over a Galois ring are constructed.

Keywords: *digit-injective matrix, DI-matrix, permutation, Galois ring.*

Bondarenko L. N. **PROPERTIES OF STATISTICS VAR ON GROUP OF PERMUTATIONS.** Some properties of statistics var defining the number of various symbols in a word obtained by component addition mod n of permutation of degree n with a fixed permutation (a key) are considered.

Keywords: *permutation, statistics, generating polynomial, permanent, circulant.*

Bylkov D. N. **THE SECOND COORDINATE SEQUENCE OF A LINEAR RECURRENCE OF MAXIMUM PERIOD OVER RING \mathbb{Z}_8 .** The analytical structure of the second coordinate in a linear recurrence sequence over ring \mathbb{Z}_8 is described. The lower bound of its rank (linear complexity) is specified. The class of polynomials and recurrences of the maximum period with the highest possible rank is found.

Keywords: *linear recurring sequence, coordinate sequence, rank, analytical structure.*

Volgin A. V. **AN IMPROVED ESTIMATE FOR THE CONVERGENCE RATE IN THE MULTIDIMENSIONAL CENTRAL LIMIT THEOREM.** An improvement is offered for the estimate of the convergence rate in the multidimensional central limit theorem. The value of the offered estimate obviously depends on the dimension of the random vectors.

Keywords: *multidimensional central limit theorem, rate of convergence, locally dependent random vectors.*

Geut Kr. L., Titov S. S. **ON POLYQUADRATIC EXTENSION OF BINARY FIELDS.** The paper is devoted to generation of irreducible polynomials of degree 2^n by using polyquadratic field extension of $\text{GF}(2)$. Full binary tree of these polynomials is constructed. Some properties of such extension are formulated.

Keywords: *irreducible polynomial, polyquadratic extension, the trace of polynomial.*

Zaets M. V. **CLASSES OF POLYNOMIAL AND VARIATIVE COORDINATE POLYNOMIAL FUNCTIONS OVER GALOIS RING.** A new class of functions over Galois ring $R = \text{GR}(q^m, p^m)$ named functions with the variative coordinate polynomiality (VCP-functions) is introduced. The relation between this class and the class of polynomial functions over R is considered. An upper bound for the amount of such functions is presented, and sufficient conditions for a VCP-function not to be a polynomial are given.

Keywords: *polynomial functions, Galois ring, coordinate set, VCP-functions.*

Kolomeec N. A. **AN AFFINE PROPERTY OF BOOLEAN FUNCTIONS ON SUBSPACES AND THEIR SHIFTS.** Let a Boolean function in n variables be affine on an affine subspace of dimension $\lceil n/2 \rceil$ if and only if f is affine on any its shift. It is proved that algebraic degree of f can be more than 2 only if there is no affine subspace of

dimension $\lceil n/2 \rceil$ that f is affine on it.

Keywords: *Boolean functions, bent functions, quadratic functions.*

Kurgansky O. M. **ON ALGORITHMIC AND TOPOLOGICAL PROPERTIES OF ORBITS FOR PIECEWISE-AFFINE MAPPINGS.** The open reachability problem for one dimensional piecewise-affine mappings with two intervals (2-PAM) is considered. Some decidability results following from the specific topological properties of reachable states of the 2-PAM's are given.

Keywords: *piecewise-affine mapping, reachability problem.*

Mironenko O. L. **STATISTICAL INDEPENDENCE OF GENERAL SUPERPOSITION OF BOOLEAN FUNCTIONS.** The sufficient conditions are proved for a superposition of some Boolean functions to be statistically independent on the subset of variables.

Keywords: *Boolean functions superposition, statistical independence.*

Filyuzin S. Y. **ALGEBRAIC IMMUNITY UPPER BOUND FOR SOME DILLON'S BENT FUNCTIONS.** An upper bound for the algebraic immunity of some Dillon's bent functions is obtained. It is shown that for $k = 2, 3, \dots, 8$ the degree for Tu and Deng's function in 2^k variables used in the Dillon's method for constructing bent functions of the maximum algebraic immunity equals $k - 1$.

Keywords: *Boolean function, nonlinearity, bent function, algebraic immunity.*

Fomichev V. M. **EQUIVALENCE OF PRIMITIVE SETS.** Equivalence of primitive sets of natural numbers is investigated in connection with the diophantine Frobenius problem. The equivalence is used to simplify calculations of Frobenius number $g(a_1, \dots, a_k)$ and all numbers that are not contained in the additive semigroup generated by the set $\{a_1, \dots, a_k\}$.

Keywords: *Frobenius's function, primitive set, additive semigroups of numbers.*

Frolova A. A. **AN ITERATIVE CONSTRUCTION OF ALMOST PERFECT NONLINEAR FUNCTIONS.** Vectorial Boolean functions F and G are equivalent if $\forall a \neq 0 \forall b [\exists x(F(x) \oplus F(x \oplus a) = b) \Leftrightarrow \exists x(G(x) \oplus G(x \oplus a) = b)]$. It is proved that every class of equivalent almost perfect nonlinear (APN) functions in n variables contains 2^{2n} different functions. An iterative procedure is proposed for constructing APN functions in $n + 1$ variables from two APN and two Boolean functions in n variables satisfying some conditions. Computer experiment show that among functions in small variables there are many functions satisfying these conditions.

Keywords: *vectorial Boolean function, APN function, γ -equivalence, iterative construction.*

Cheremushkin A. V. **ON A NONLINEARITY DEGREE DEFINITION FOR A DISCRETE FUNCTION ON A CYCLIC GROUP.** An additive approach is proposed to the definition of the nonlinearity degree of a discrete function on a cyclic group. For elementary abelian groups, this notion is equivalent to ordinary "multiplicative" one. For polynomial functions on a ring of integers mod p^n , this notion is equivalent to minimal degree of a polynomial. It is shown that the nonlinearity degree is a finite number if and only if the order of the group is a power of a prime. An upper bound for the nonlinearity degree of functions on a cyclic group of order p^n is given.

Keywords: *nonlinearity degree, discrete functions.*

Sholomov L. A. **AN ECONOMICAL REPRESENTATION OF UNDERDETERMINED DATA AND SUPERIMPOSED CODES.** For underdetermined data, economical representations making it possible to reconstruct the initial data are proposed. A connection between representations and superimposed codes is found, and bounds for representations length are obtained.

Keywords: *underdetermined data representation, superimposed code, cover-free matrix.*

SECTION 2

Vitkup V. A. **ON THE REPRESENTATION OF S-BOXES IN BLOCK CIPHERS.** A known method of S-boxes partition applied against side-channel attacks is considered. Nowadays, necessary partitions are found for all the affine equivalence classes except one. In the paper, it is proved that S-boxes of this class do not have admissible partition.

Keywords: *S-box, vectorial Boolean function, affine equivalence.*

Kaluzhin A. K., Chizhov I. V. **ALGORITHM FOR RECOVERING PLAINTEXT FROM CIPHERTEXT IN MCELIECE CRYPTOSYSTEM.** An attack on McEliece cryptosystem is considered. In it a plaintext is recovered from a ciphertext by solving the encryption equation. The solution is get in two steps: finding the error vector and solving the system of linear equations. For finding the error vector, the Bernstein — Lange — Peters's algorithm is used together with some optimization techniques. The complexity of the offered attack on the cryptosystem based on Goppa (1024, 524, 50)-code equals $2^{60,1}$ bit operations that is 27,5% less than by means of Bernstein — Lange — Peters's algorithm itself.

Keywords: *McEliece's cryptosystem, nonstructural attacks, Bernstein — Lange — Peters's algorithm.*

Karpunin G. A. **ON PROBABILITY CHARACTERISTICS OF RANDOM GRAPHS GENERATED BY ALGORITHMS FOR FINDING HASH FUNCTION COLLISIONS.** In the paper, a graph model of some algorithms for finding SHA-1 and RIPEMD collisions is described, and under the described model, an exact formula for calculating average complexity of these algorithms is given.

Keywords: *cryptographic hash functions, collisions, random graphs.*

Katerinskiy D. A. **ABOUT INVERTIBILITY FINITE AUTOMATA WITH FINITE DELAY.** Experimental estimates are obtained for the proportion of invertible, weakly invertible and strong invertible finite automata with finite delay. The estimates show that the proportion of the invertible automata is small (about 3%) for automata with near numbers of states and output symbols and is large (over 80%) for automata with the number of output symbols being 4 times more than the number of input symbols and 2 times more than the number of states.

Keywords: *finite automata, weakly invertibility, invertibility, analysis of invertibility, synthesis of inverse automata, proportion of invertible automata.*

Kovalev D. S. **FPGA IMPLEMENTATION OF FAPKC SYMMETRIC EQUIVALENT.** FPGA implementation of the FAPKC symmetric equivalent (called FASKC) is presented. The throughput/area comparison of the FASKC with the other finite automata cryptosystems is made. The FPGA implementation comparison of the FASKC, AES and other contemporary block ciphers is given.

Keywords: *non-linear automaton, invertible with delay automaton, finite automata cryptosystem, FAPKC, FASKC, PLD, FPGA, VHDL.*

Koreneva A. M. **BLOCK CIPHERS BASED ON TWO FEEDBACK SHIFT REGISTERS.** The conditions of providing bijectivity and involutivity properties are obtained for the block encryption algorithms which are based on a shift register with two feedbacks over the space of binary vectors. An example of a block encryption algorithm of this kind is constructed. The algorithm is based on a shift register of length 4.

Keywords: *bijectivity, iterative symmetric block ciphers, involutivity of encryption algorithm, shift registers*

Medvedev N. V., Titov S. S. **CONSTRUCTIONS OF IDEAL SECRET SHARING SCHEMES.** Linear homogeneous ideal secret sharing schemes are considered. The construction of such schemes is given over any field $GF(q)$. By adding participants it is shown that such schemes are reduced to schemes on projective spaces.

Keywords: *homogeneous secret sharing schemes, matroids, Reed — Muller code*

Medvedeva N. V., Titov S. S. **ON NON-MINIMAL PERFECT CIPHERS.** An analogue of Shannon's theorem is proved for non-endomorphic ciphers.

Keywords: *perfect ciphers, non-endomorphic ciphers, maximal ciphers, non-minimal ciphers.*

Pestunov A. I. **ON RELATIONS BETWEEN THE BASIC NOTIONS OF DIFFERENTIAL CRYPTANALYSIS.** Some problems and inconsistencies in terminology related to the differential cryptanalysis of iterative block ciphers are considered. A set of definitions is suggested to solve these problems and to form a system of unified notions with no contradictions. By using the suggested definitions it is shown that the truncated characteristic is the most general notion: differential, truncated differential and characteristic are in fact particular cases of the truncated characteristic.

Keywords: *terminology, differential cryptanalysis, block cipher, characteristic.*

Chizhov I. V., Borodin M. A. **THE FAILURE OF MCELIECE PKC BASED ON REED — MULLER CODES.** This paper describes new algorithm for breaking McEliece cryptosystem, being built on Reed — Muller binary code $RM(r, m)$. The algorithm calculates the private key from the public key using $O(n^d + n^4 \log_2 n)$ bit operations, where $n = 2^m, d = (r, m - 1)$. In case of limited d , the attack has a polynomial complexity. Practical results of implementation show that McEliece cryptosystems, based on the Reed — Muller binary code of length $n = 65526$ bits, can be broken in less than 7 hours on a personal computer.

Keywords: *McEliece cryptosystem, Reed — Muller code, polynomial attack.*

SECTION 3

Alekhina M. A., Barsukova O. U. **ABOUT UNRELIABILITY BOUNDS FOR CIRCUIT WITH INVERSE FAULTS AND FUNCTIONAL ELEMENT BREAKDOWNS.** The realization of Boolean functions by circuits of unreliable functional elements is considered in an arbitrary complete basis. It's supposed that all circuit elements are independently of each other prone to faults of two types: output inverse faults and element breakdowns. Upper and lower asymptotical bounds of circuit unreliability are presented.

Keywords: *Boolean functions, functional element, circuit, unreliability of circuit, output inverse faults, element breakdowns.*

Anisenya N. I., Stefantsov D. A., Torgaeva T. A. **THE BLACKBOX SERVICE FOR HOSTING CAPTURE THE FLAG COMPUTER SECURITY COMPETITIONS.** The BlackBox system developed by the Tomsk State University team SiBears for hosting the task-based Capture the Flag competitions in computer security is introduced. The functionality of the system is described along with the peculiarities of its development and administration. The proposals about the future development are made.

Keywords: *SiBears, BlackBox, CTF.*

Vasin A. V. **ABOUT BASES WHOSE UNRELIABILITY COEFFICIENT EQUALS 1.** Circuits composed of unreliable functional elements in a complete finite basis B are considered. It is assumed that all elements are independently of each other subjected to inverse failures at the outputs with the probability ε ($\varepsilon \in (0, 1/2)$). In the paper, a set G of Boolean functions is found, and it is proved that if $B \cap G \neq \emptyset$, then almost all Boolean functions are realized in basis B by asymptotically optimal on reliability circuits with unreliability ε under $\varepsilon \rightarrow 0$.

Keywords: *unreliable functional gates, circuits asymptotically optimal with respect to reliability, inverse failures on outputs of gates.*

Devyanin P. N. **CORRECTNESS OF STATE TRANSFORMATION RULES IN MROSL DP-MODEL.** Conditions and results of application are analysed for state transformation rules in mandatory entity-role security model of access and information flows control in OS of Linux set (MROSL DP-model). The correctness of the rules is considered with regard to requirements of mandatory access control (MAC), mandatory integrity control (MIC) and role-based access control (RBAC).

Keywords: *computer security, formal model, access control.*

Zaytsev G. Yu., Potapkin A. I., Stefantsov D. A. **MODIFICATION OF COMPILED APPLICATIONS FOR THE ANDROID PLATFORM BY MEANS OF ASPECT-ORIENTED PROGRAMMING.** The tool for modification of compiled applications for the Android platform by means of aspect-oriented programming is presented. It is based on the Aspect-Oriented Programming paradigm, is implemented with the ASMDEX library, and performs the weaving of the program and the aspects in two passes. The language for implementation of the aspects is Java with special annotations encapsulating the necessary meta-information.

Keywords: *aspect-oriented programming, Android, Dalvik.*

Kolegov D. N., Tkachenko N. O., Chernov D. V. **DEVELOPMENT AND IMPLEMENTATION OF MANDATORY ACCESS CONTROL MECHANISMS IN DBMS MYSQL.** The paper is devoted to development and implementation of mandatory access control mechanisms for DBMS MySQL based on discretionary access policy. A formal security model is proposed for multilevel security mandatory access policy. It is implemented in MySQL core reference monitor enabling to protect DBMS against prohibited information and match security requirements for trusted computer systems.

Keywords: *computer security, access control, information flows, formal security model.*

Shcherba E. V., Volkov D. A. **DEVELOPMENT OF A DDOS-ATTACK DETECTION SYSTEM USING QUEUING THEORY.** A specialized system architecture is proposed for DDoS attack detection. It is based on the evaluation of the packet loss probability and the theory of queuing networks.

Keywords: *network attacks detection, denial of service, DDoS.*

SECTION 4

Abrosimov M. B., Modenova O. V. **ABOUT THE LOWER BOUNDS FOR THE NUMBER OF ADDITIONAL ARCS IN A MINIMAL VERTEX 1-EXTENSION OF ORIENTED PATH.** A graph G^* with $n + k$ vertices is vertex k -extension of a graph G if every graph obtained by removing any k vertices from G^* contains G ; it is called minimal vertex k -extension of G if it has the least number of arcs among all vertex k -extensions of graph G with $n + k$ vertices. A lower bound for the number of additional arcs in minimal vertex 1-extension of an oriented path is given.

Keywords: *graph, minimal vertex extension, fault tolerance.*

Batueva T. **PROPERTIES OF GENE NETWORKS WITH THRESHOLD FUNCTIONS.** An algorithm for finding all fixed points of the state graph of a circulant type gene network transformed by a Boolean function is given. All sources of the state graph of a gene network transformed by a threshold Boolean function in k variables with a single value 1 are described. In case $k = 3$ all circles of the state graph are described too, and the length of the maximum chain in it is calculated.

Keywords: *gene network, directed graph, threshold functions, state graph of mapping, fixed point, source of state graph.*

Bondarenko P. P. **ON THE UPPER BOUND FOR THE NUMBER OF ADDITIONAL EDGES IN MINIMAL VERTEX EXTENSIONS OF COLORED CIRCLES.** An upper bound for the number of additional edges in the minimum vertex 1-extensions of cycles with the vertices of two types and a general construction of one of such extensions are given.

Keywords: *graph, circle, minimal extension, fault-tolerance.*

Evdokimov A. A., Kochemazov S. E., Otpushennikov I. V., Semenov A. A. **DYNAMICAL PROPERTIES OF SOME DISCRETE AUTOMATON MAPPINGS DEFINED BY RANDOM GRAPHS.** In this report, the results of computational analysis are presented for problems of searching fixed points and cycles of some discrete mappings, that are used to model the behaviour of systems with many interconnecting agents and are defined by random graphs generated according to known models (G_{np} -graphs, the Watts — Strogatz model).

Keywords: *random graphs, gene networks, discrete automaton mappings, SAT.*

Zharkova A. V. **ON BRANCHING AND IMMEDIATE PREDECESSORS OF THE STATES IN FINITE DYNAMIC SYSTEM OF ALL POSSIBLE ORIENTATIONS OF A GRAPH.** Branching and immediate predecessors of the states in the finite dynamic system of all possible orientations of a given graph are found. Evolutionary function of the system transforms digraphs by reorientation of all arcs entering the sinks. The inaccessibility property is defined for a state in this dynamic system.

Keywords: *finite dynamic system, graph, graph orientation, branching, inaccessibility, immediate predecessor.*

Komarov D. D. **MINIMAL EDGE EXTENSIONS OF SPECIAL TYPE PALM TREES.** Minimal edge 1-extensions of 2-leaf palm trees are described.

Keywords: *extensions of graphs, palm trees.*

Kornienko A. S. **FUNCTIONAL GRAPH TREES FOR CIRCULANTS WITH LINEAR BOOLEAN FUNCTIONS AT THE VERTICES.** The functional graph of a discrete dynamic system being a model of regulatory gene network circuit is de-

defined as the graph of the transformation $A_{f,2} : F_2^n \rightarrow F_2^n$ where $A_{f,2}(v_0, v_1, \dots, v_{n-1}) = (u_0, u_1, \dots, u_{n-1})$, $u_i = v_{i-1} + v_i + v_{i+1}$, $i = 0, 1, \dots, n-1$, $v_{-1} = v_{n-1}$, $v_n = v_0$. The structure of this graph is completely described.

Keywords: *discrete dynamical system, circulant, gene network, regulatory circuit, functional graph.*

Kyazhin S. N. ON LOCAL PRIMITIVENESS OF GRAPHS AND NONNEGATIVE MATRICES. Cryptographic generators constructed of control and generating blocks are investigated. Essential dependence of block elements on all signs of generator initial state is the useful property of such generators. The notion of a local primitiveness for a nonnegative matrix or graph is introduced to study such dependences. The conditions for matrix local primitiveness are obtained. A relation between the local primitiveness characteristics of matrices (graphs) of particular classes and parameters of generators is established.

Keywords: *exponent, local exponent, primitive matrix, primitive graph, local primitiveness.*

Nazhmidenova A. M. THE DISCRETE DYNAMIC SYSTEM ON A DOUBLE CIRCULANT WITH DIFFERENT FUNCTIONS AT THE VERTICES. The structure of the functional graph is studied for a discrete dynamic system consisting of two circulants $G_{n,k}$ with different orientations and functionings and with the corresponding vertices being conjugate. The recurrent relation for the number of fixed points is obtained, and the asymptotic behaviour of this number is described. In the case $k = 2$ the theorems characterizing structural properties, fixed points, pendant vertices and cycles of length 2 of the functional graphs are proved. In particular, the explicit formulas for the number of fixed points and pendant vertices are found.

Keywords: *gene network, discrete model, regulatory loop, circulant, functional graph, cycles, fixed points, pendant vertices.*

Osipov D. U. ON T-IRREDUCIBLE EXTENSIONS OF STARLIKE TREES. T-irreducible extension is a kind of the optimal extension of a graph. In the paper, all nonisomorphic T-irreducible extensions are constructed for starlike trees with paths of one and the same length.

Keywords: *graph, T-irreducible extension, starlike trees.*

Salii V. N. ON THE ORDERED SET OF CONNECTED PARTS OF A POLYGONAL GRAPH. Polygonal graphs, whose the ordered set of abstract connected parts is a lattice, are characterized.

Keywords: *polygonal graph, linear graph, binary vector, duality, ordered set, lattice.*

Tokareva N. N. SIMPLE PROOF FOR THE STRONG REGULARITY OF THE CAYLEY GRAPH OF BENT FUNCTION. A simple proof is presented for the known result about the strong regularity of Cayley graph of a bent function.

Keywords: *bent functions, strongly regular graphs.*

Tsitsiashvili G. Sh., Osipova M. A., Losev A. S. ASYMPTOTICS OF CONNECTIVITY PROBABILITIES FOR PAIRS OF GRAPH NODES. For graphs with low reliable arcs, asymptotics of probabilities for connectivities between all pairs of nodes are constructed. Parameters of these asymptotics are characteristics of shortest paths in the graph. To calculate these characteristics, some modifications of classical algorithms are developed. On the base of these results, numerical experiment is realized. This experiment

demonstrates advantages of suggested algorithms.

Keywords: *shortest path, connectivity probability, computational complexity.*

SECTION 5

Азибалов Г. П., Липский В. Б., Панкратова И. А. **КРИПТОГРАФИЧЕСКОЕ РАСШИРЕНИЕ РУССКОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ.** Представлено расширение русского языка программирования ЛЯПАС, получившее название ЛЯПАС-Т и заключающееся в увеличении длины операндов и расширении множества элементарных операций над ними. Необходимость в нём продиктована, в первую очередь, потребностями страны в доверенных и эффективных программной и аппаратной реализациях современных криптографических алгоритмов в безопасных компьютерных системах логического управления критически важными объектами, такими, как космические системы, энергетические установки, ядерное оружие, подводные лодки, беспилотники и т. п. Представлен также компилятор ЛЯПАСа-Т, генерирующий его загрузочный модуль для операционной системы Linux.

Ключевые слова: *русский язык программирования, криптографическое расширение, ЛЯПАС-Т, компилятор.*

Азибалов Г. П., Липский В. Б., Панкратова И. А. **ПРОЕКТ АППАРАТНОЙ РЕАЛИЗАЦИИ РУССКОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ.** Представлен проект процессора, реализующего ЛЯПАС-Т аппаратно, и предпроцессора, конвертирующего программы на ЛЯПАСе-Т в исполняемый код процессора. Сообщается о процессоре для подмножества ЛЯПАСа-Т без подпрограмм, операций над комплексами и длинных операндов, описанном на VHDL, протестированном средствами компьютерного моделирования и реализованном на ПЛИС с помощью системы автоматизированного проектирования.

Ключевые слова: *русский язык программирования, ЛЯПАС-Т, аппаратная реализация, предпроцессор.*

Брославский О. В. **AES НА ЛЯПАСЕ.** Представлены программы на языке ЛЯПАС, реализующие симметричный блочный алгоритм шифрования AES и расширение ключа для него.

Ключевые слова: *AES, ЛЯПАС.*

SECTION 6

Anashkina N. V. **ABOUT POSSIBILITY OF REDUCTION OF SORT OUT IN BALASH'S ALGORITHM.** An optimization of Balash's algorithm using particular feature of geometric structure of deadlock point's environs is presented.

Keywords: *Balash's algorithm, discrepancy, deadlock point.*

Arbuzov D. S., Tuktarova L. I. **ANALYSIS OF SOME ALGORITHMS FOR SMOOTH INTEGERS RECOGNITION.** The experimental comparison of three sieving algorithms by run time and memory amount is presented.

Keywords: *smooth numbers, sieving, Bernstein algorithm.*

Bulavintsev V. G., Semenov A. A. **GPU-BASED IMPLEMENTATION OF DPLL ALGORITHM WITH LIMITED NON-CHRONOLOGICAL BACKTRACKING.** A new GPU-based SAT solver named ngsat is presented. The solver employs DPLL

algorithm with limited version of non-chronological backtracking without Clause Learning. Some new techniques are developed and applied to increase the effectiveness of DPLL algorithm on SIMD. Ngsat's performance is demonstrated in application to the problems of search for pairs of orthogonal Latin squares.

Keywords: GPU, DPLL algorithm, SAT, parallel computer architectures, CUDA, SIMD.

Bykova V. V. ASYMPTOTIC SOLUTION OF THE RECURRENCE RELATIONS IN THE ANALYSIS OF SPLITTING ALGORITHMS FOR SAT. The traditional technique for analysis of splitting algorithms for SAT problem is considered. A theorem establishing the upper bounds for execution time of algorithms in the case of balanced splitting is offered.

Keywords: splitting algorithms, computational complexity.

Zhukov K. D., Rybakov A. S. ON SOLVING BIG SYSTEMS OF CONGRUENCES. Let S be a finite set of positive integers such that almost all its elements are pairwise coprime. An algorithm is presented for finding all elements $s \in S$, such that $(s, s') > 1$ for an element $s' \in S$, $s' \neq s$. The algorithm allows to reduce any system of polynomial congruences to a number of systems with coprime moduli.

Keywords: coprime base, gcd, merge gcd, gcd tree.

Klimina A. S. OPTIMIZATION OF POLLARD'S $(p-1)$ -ALGORITHM. The article contains criteria for choice of parameters and a method for optimization of the Pollard's $(p-1)$ -algorithm.

Keywords: Pollard's $(p-1)$ -algorithm, integer factorization.

Kuznetsova A. S., Kuznetsov A. A., Safonov K. V. A PARALLEL ALGORITHM FOR COMPUTATION OF GROWTH FUNCTIONS IN THE FINITE TWO-GENERATOR GROUPS OF PERIOD 5. A parallel version of the algorithm for computation of growth functions in the finite two-generator groups of period 5 is presented.

Keywords: the growth function of the group, the Cayley diameter, a parallel algorithm.

Pottosin Yu. V., Kardash S. N. PIPELINING OF COMBINATIONAL CIRCUITS. The problem is set to divide a given multilevel combinational circuit into a given number of cascades with registers providing pipeline-wise development of incoming signals. To solve this problem a model based on representation of combinational circuit in the form of digraph is used.

Keywords: combinational circuit, pipelining, directed graph.

Ryabokon D. V. ALGORITHM FOR SEARCHING PROHIBITIONS OF BOOLEAN FUNCTIONS. An algorithm for search of prohibitions of Boolean function based on the branch and bound method is proposed. It allows to find a prohibition of Boolean function, a prohibition of minimum length or all prohibitions under a specified length.

Keywords: prohibition of Boolean function, de Bruijn graph.

Semenov A. A. ON THE EFFECTIVE REPRESENTATION OF DISJUNCTIVE NORMAL FORMS BY DIAGRAMS OF A SPECIAL KIND. For an arbitrary disjunctive normal form of a Boolean function, a disjunctive diagram representation is proposed. This kind of diagrams is constructed in a polynomial time and can be used to reduce the size of conflict databases produced during non-chronological DPLL derivation.

Keywords: decision diagrams, BDD, ZDD, disjunctive diagrams.

Usatyuk V. S. IMPLEMENTATION OF THE PARALLEL SHORTEST VECTOR ENUMERATION IN THE BLOCK KORKIN — ZOLOTAREV METHOD.

A parallel CPU implementation of Kannan algorithm is presented for solving shortest vector problem in block Korkin — Zolotarev lattice reduction method. The implementation is based on Native POSIX Thread Library and shows the linear decrease of runtime with the number of threads.

Keywords: *shortest vector problem, SVP, block Korkin — Zolotarev, BKZ, lattices, parallel algorithms.*

Chernyak R. I. **PARALLELIZATION OF THE DECODING ALGORITHM IN VIDEO COMPRESSION STANDARD H.265/HEVC.** A method for parallel implementation of the decoder in the video compression standard H.265/HEVC is proposed. The effectiveness of the method is proved theoretically and shown experimentally.

Keywords: *H.265/HEVC, digital video compression, HEVC decoder parallelization.*

Shangin R. E. **EXACT ALGORITHM FOR SOLVING SPECIAL CASE OF DISCRETE WEBER PROBLEM.** An algorithm reasonably solving Weber problem for n -sequentially connected chain and finite set of points of location is described. The algorithm is compared with an integer linear programming algorithm realized in IBM ILOG CPLEX.

Keywords: *Weber problem, n -sequentially connected chain, dynamic programming, exact algorithm, quasi-polynomial algorithm.*