МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Приложение

№ 13 **Сентябрь 2020**

Зарегистрирован в Федеральной службе по надзору
в сфере связи и массовых коммуникаций

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

ТРУДЫ
Всероссийской конференции
«XIX Сибирская научная школа-семинар с международным участием
“Компьютерная безопасность и криптография” — SIBECRYPT’20»

УЧРЕДИТЕЛЬ
Томский государственный университет

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

Теоретические основы прикладной дискретной математики
Дискретные функции
Математические методы криптографии
Математические основы компьютерной безопасности
Прикладная теория кодирования, автоматов и графов
Математические основы информатики и программирования
Вычислительные методы в дискретной математике

Редактор *Н. И. Шидловская*
Верстка *И. А. Панкратовой*

Подписано к печати 15.08.2020. Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 17,56. Тираж 300 экз.
Заказ № 4381. Цена свободная. Дата выхода в свет 27.08.2020.

Отпечатано на оборудовании
Издательского Дома Томского государственного университета
634050, г. Томск, пр. Ленина, 36
Тел.: 8(3822)53-15-28, 52-98-49

СОДЕРЖАНИЕ

Секция 1

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

Секция 2

ДИСКРЕТНЫЕ ФУНКЦИИ

Секция 3

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

Секция 4

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

Секция 5

ПРИКЛАДНАЯ ТЕОРИЯ КОДИРОВАНИЯ, АВТОМАТОВ И ГРАФОВ

Секция 6

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

Секция 7

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

Секция 1

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

УДК 519.151, 519.725, 519.165 DOI 10.17223/2226308X/13/1

ПРЕЛОМЛЯЮЩИЕ БИЕКЦИИ В ТРОЙКАХ ШТЕЙНЕРА

М. В. Ведунова, К. Л. Геут, А. О. Игнатова, С. С. Титов

Исследованы преломляющие биекции в тройках Штейнера, применяемые при построении матроидов и схем разделения секрета. Под преломляющими понимаются отображения F квазигруппы в себя, удовлетворяющие условию $F(x * y) \neq \neq F(x) * F(y)$ при любых $x \neq y$. Предложены преломляющие биекции квазигрупп Штейнера с $N = 9, 13$ и $2^n - 1$ элементами при нечётных n, не делящихся на три, а также необходимые условия существования APN-биекций в $\mathrm{GF}(2^n)$. При помощи наборов преломляющих биекций построены матроиды, являющиеся контрпримерами к гипотезе, что каждый однородный матроид определяет некоторую блок-схему.

Ключевые слова: *преломляющие биекции, квазигруппы Штейнера, матроиды.*

Исследуются биекции квазигрупп, которые по аналогии с геометрическими преобразованиями, не переводящими никакую прямую в другую прямую, названы преломляющими. Преломляющие биекции применимы при построении APN-функций и как контрпример к гипотезе в теории схем разделения секрета: оказалось, что не каждый однородный матроид определяет некоторую блок-схему [1]. APN-биекции неоднократно изучались, в том числе вопросу существования взаимно однозначных APN-функций от чётного числа переменных посвящены работы [2, 3]. Итеративные конструкции APN-функций исследованы в [4].

Построение систем троек Штейнера S, S', S'' на множестве $G = \{1, 2, \ldots, N\}$, таких, что никакие две из них не содержат ни одной общей тройки [5], выглядит следующим образом: тройка $\{u, v, w\}$ преобразуется в тройку $\{f(u), f(v), f(w)\}$, где

1) $\{f(u), f(v), f(w)\} \in S'$;
2) $\{g(u), g(v), g(w)\} \in S''$.

Утверждение 1. Если существуют три биекции $F(x) = f(x)$, $F(x) = g(x)$ и $F(x) = fg^{-1}(x)$ квазигруппы Штейнера $(S, *)$, являющиеся преломляющими, т. е. удовлетворяющие условию $F(x * y) \neq F(x) * F(y)$ при любых $x \neq y$, то соответствующие им системы S, S', S'' не содержат ни одной общей тройки.

Системы S, S', S'' образуются в результате применения преломляющих биекций к стандартным тройкам Штейнера [5].

Рассмотрим на множестве $E = \{a, b, c\} \cup G = \{a, b, c, 1, 2, \ldots, N\}$ семейство $\mathcal{H}$ четырёхэлементных подмножеств четырёх видов:

1) $H = \{a, u, v, w\}$, где $\{u, v, w\} \in S$;
2) $H' = \{b, i, j, k\}$, где $\{i, j, k\} \in S'$;
3) $H'' = \{c, x, y, z\}$, где $\{x, y, z\} \in S''$;
4) $H''' = \{a, b, c, t\}$, где $t \in G$.

Утверждение 2. Семейство $\mathcal{H}$ удовлетворяет аксиомам гиперплоскостей матроида, оно не является семейством блоков никакой блок-схемы, причём двойственный матроид — однородный с мощностью циклов, равной $n = N - 1$.

Таким образом описывается конструкция контрпримера. На данный момент рассмотрены линейные системы с $N = 2^n - 1$, а также системы с $N = 9$ и нелинейные с $N = 13$. Оказалось, что при $N = 7$ таких S, S', S'' нет, при $N = 9$, 13 и 31 такие системы существуют, при $N = 15$ — пока неизвестно.

Системы троек Штейнера на N элементах существуют тогда и только тогда, когда $N = 1 \pmod 6$ или $N = 3 \pmod 6$ [5].

Утверждение 3. При $N = 9$ существуют системы троек Штейнера S_9, S'_9, S''_9 без общих троек.

Доказательство и построение преобразований, преломляющих прямые, производится методом решения задачи блокировки прямых [6], поскольку система троек S_9 есть семейство прямых на аффинной плоскости порядка три.

Утверждение 4. Биекции

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 13 & 1 & 2 & 4 & 8 & 12 & 11 & 10 & 7 & 6 & 9 & 5 & 3 \end{pmatrix},$$

$$g = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 8 & 6 & 10 & 2 & 11 & 3 & 1 & 12 & 4 & 5 & 7 & 13 & 9 \end{pmatrix},$$

$$f(g^{-1}) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 12 & 7 & 4 & 9 & 1 & 8 & 5 & 3 & 11 & 2 & 13 & 10 & 6 \end{pmatrix},$$

преобразующие стандартные тройки S_{13} первого типа [5], являются преломляющими.

Пусть $f(u \oplus v) \neq f(u) \oplus f(v)$ для любых различных ненулевых u и v из F_2^n, при этом в линейных системах троек Штейнера $F(0) = 0$. Это равносильно тому, что f преобразует любое двумерное подпространство пространства F_2^n в четырёхэлементное подмножество, содержащее нуль, но не являющееся двумерным подпространством, и является необходимым условием для APN-биекции f, сохраняющей нуль.

Если A и B — линейные невырожденные преобразования пространства F_2^n, то для суперпозиции биекций AfB имеем $A(f(B(u \oplus v))) = A(f(Bu \oplus Bv)) \neq A(f(Bu) \oplus \oplus f(Bv)) = Af(Bu) \oplus Af(Bv)$, $A(f(B(0))) = 0$, то есть AfB тоже обладает этим свойством.

Нетрудно проверить, что в $\mathrm{GF}(2^n)$ при $n = 3$ (т. е. $N = 7$) суперпозиция любых двух преломляющих биекций не является преломляющей. Поэтому представляет интерес случай $n > 3$.

Утверждение 5. Функция $F(u) = u^{-3}$ не сохраняет двумерные линейные подпространства в $\mathrm{GF}(2^n)$ при нечётном n, т. е. является преломляющей, тогда и только тогда, когда $\mathrm{GF}(2^n)$ не содержит $\mathrm{GF}(2^3)$, то есть n не делится на 3.

Утверждение 6. При нечётном n, где n не делится на 3, функции $f(u) = u^3$ и $F(u) = u^{-3}$ являются преломляющими вместе с функцией $g(u) = u^{-1}$.

Это утверждение вместе с утверждениями 1 и 2 позволяет строить однородные матроиды мощности $2^n + 2$, удобные для использования в схемах разделения секрета, не сводящиеся к блок-схемам, для нечётных $n \geqslant 5$ при помощи систем линейных троек Штейнера с квазигрупповой операцией $\oplus$.

Однако поскольку APN-биекции, сохраняющие нуль, являются преломляющими, стоит отметить отрицательный результат.

Утверждение 7. Биекция $F(u) = u^{-3}$ не является APN-функцией.

ЛИТЕРАТУРА

1. *Медведев Н. В., Титов С. С.* Об однородных матроидах и блок-схемах // Прикладная дискретная математика. Приложение. 2017. № 10. С. 21–23.
2. *Идрисова В. А.* Векторные 2-в-1 функции как подфункции взаимно однозначных APN-функций // Прикладная дискретная математика. Приложение. 2018. № 11. С. 39–41.
3. *Виткуп В. А.* О специальном подклассе векторных булевых функций и проблеме существования APN-перестановок // Прикладная дискретная математика. Приложение. 2016. № 9. С. 19–21.
4. *Фролова А. А.* Итеративная конструкция APN-функций // Прикладная дискретная математика. Приложение. 2013. № 6. С. 24–25.
5. *Холл М.* Комбинаторика: пер. с англ. М.: Мир, 1970. 424 с.
6. *Ведунова М. В., Игнатова А. О., Геут К. Л.* Блокировка линейных многообразий и тройки Штейнера // Прикладная дискретная математика. Приложение. 2019. № 12. С. 93–95.

УДК 519.151, 519.725, 519.165 DOI 10.17223/2226308X/13/2

ОБ ОДНОРОДНЫХ МАТРОИДАХ, СООТВЕТСТВУЮЩИХ БЛОК-СХЕМАМ

Н. В. Медведев, С. С. Титов

Исследуются взаимосвязи однородных матроидов и блок-схем. Эта задача связана с изучением структур доступа идеальных совершенных схем разделения секрета. Под однородностью матроида понимается одинаковая мощность его циклов, при этом, возможно, не все подмножества этой мощности являются циклами. Для мощности циклов пять доказано, что однородный связный разделяющий матроид является равномерным. При этом если матроид связный и разделяющий, то двойственный ему матроид будет простым. Доказано, что если каждый цикл однородного разделяющего связного матроида является его гиперплоскостью, то ему соответствует блок-схема.

Ключевые слова: *однородные матроиды, схемы разделения секрета, блок-схемы, циклы.*

На множестве M определён матроид, если некоторые его подмножества названы независимыми (остальные — зависимыми), причём удовлетворяются аксиомы матроида; так, в терминах циклов — минимальных (по включению) зависимых подмножеств из M — аксиом всего две: 1) нет цикла в цикле, т. е. если C, D — циклы и $C \subseteq D$, то $C = D$; 2) если $C_1 \neq C_2$ — циклы и $x \in C_1 \cap C_2$, то $C_1 \cup C_2 \setminus \{x\}$ содержит цикл [1–4]. Матроид называется связным, если для любых двух его элементов существует содержащий их цикл. Простым (или комбинаторной геометрией) называется матроид, в котором нет одноэлементных и двухэлементных циклов. Под однородностью матроида понимается одинаковость мощностей его циклов, равная n, где, возможно, не все n-элементные множества — циклы. При этом если все n-элементные подмножества — циклы, то такой матроид называется пороговым (равномерным) [5]. Матроид является разделяющим тогда и только тогда, когда для любых $x \neq y$ существует разделяющий их цикл C, т. е. $x \notin C$, $y \in C$. Любое максимальное независимое подмножество B,

содержащееся в M, называется базой матроида M; дополнение цикла матроида — когиперплоскостью $\overline{C} = M \setminus C$.

Блок-схема $D(v, b, r, k, \lambda)$, согласно [6], — такое размещение v различных элементов по b блокам, что каждый блок содержит точно k различных элементов, каждый элемент появляется точно в r различных блоках и каждая пара различных элементов появляется в λ блоках. Блок-схема с $k = 3$ вполне естественно называется системой троек. При этом параметры должны удовлетворять аксиомам $3b = rv$, $2r = \lambda(v-1)$. Система троек с $\lambda = 1$ называется системой троек Штейнера. Условие $v \equiv 1, 3 \pmod 6$ необходимо и достаточно для существования штейнеровской системы троек.

Утверждение 1. Если каждый цикл однородного разделяющего связного матроида является его гиперплоскостью, то ему соответствует блок-схема.

Доказательство. Пусть $M = (E, \mathcal{C})$ — связный разделяющий однородный матроид с мощностью $n \geqslant 4$ циклов $C \in \mathcal{C}$, такой, что $n^* = |E| - |C| = |E| - n = 4$. Для каждой когиперплоскости H^*, т. е. гиперплоскости двойственного матроида $M^* = (E, \mathcal{H}^*)$, имеем $|H^*| = |E \setminus C| = 4$, $H^* \in \mathcal{H}^*$. Пусть $H_1^* \in \mathcal{H}^*$, $H_2^* \in \mathcal{H}^*$, $H_1^* \neq H_2^*$.

Предположим, что пересечение любых различных гиперплоскостей матроида M^* не более чем одноэлементно. Пусть $H_1^* \cap H_2^* = \{e\}$, $d \notin H_1^* \cup H_2^*$ (отметим, что ввиду $n \geqslant 4$, $n^* = 4$, $|E| = n + n^* \geqslant 8$ такой элемент d существует), тогда по второй аксиоме гиперплоскостей существует гиперплоскость $H^* \in \mathcal{H}^*$, такая, что $\{e, d\} \subset H^*$. По предположению $H^* \cap H_1^* = \{e\} = H^* \cap H_2^*$.

В силу разделимости матроида M каждый его элемент e принадлежит не менее чем двум когиперплоскостям. В самом деле: если $a \neq e$, то существует такой цикл C_a, что $a \in C_a$ и $e \notin C_a$, т. е. $e \in H_a^* = E \setminus C_a$; взяв $b \neq e$, $b \in H_a^*$, найдём цикл C_b, такой, что $b \in C_b$, $e \notin C_b$, т. е. $e \in H_b^* = E \setminus C_b$, причём $H_a^* \neq H_b^*$, так как $b \in H_a^*$, но $b \notin H_b^*$. Ввиду произвольности элемента d, в силу предположения получаем разбиение множества $E \setminus \{e\}$ множествами $H^* \setminus \{e\}$. Поскольку в качестве e можно взять любой элемент множества E, получаем блок-схему с параметрами $v = |E|$, $k = n^* = 4$, $\lambda = 1$, $r = \dfrac{v-1}{\lambda(k-1)} = \dfrac{v-1}{3}$, $b = \dfrac{vr}{k} = \dfrac{v(v-1)}{12}$. Итак, в этом случае получаем систему четвёрок Штейнера.

Каждая пара различных элементов в M^* независима, так как M^* простой, и поэтому существует единственная содержащая эту пару гиперплоскость H^*. Следовательно, ранг гиперплоскостей в M^* равен двум, а любое трёхэлементное множество — либо цикл, если оно входит в некоторую четвёрку, либо база, если не входит ни в какую четвёрку. Значит, ранг r^* матроида M^* равен трём.

Предположим, что пересечение любых двух различных гиперплоскостей матроида M^* не более чем двухэлементно. Пусть $H_1^* \cap H_2^* = \{e, f\}$, $e \neq f$. В силу связности матроида M для любых двух различных его элементов e и f существует содержащий их цикл C. Поскольку M — связный и разделяющий, то M^* простой; значит, $\{e, f\}$ — независимое множество в M^* и его можно дополнить до гиперплоскости $H_1^* = E \setminus C_1$, где C_1 — цикл в M.

Пусть $b \notin H_1^*$, тогда $\{e, f, b\}$ — независимое множество в M^* и его можно дополнить до гиперплоскости $H_2^* \neq H_1^*$, только если ранг гиперплоскостей равен трём. Следовательно, ранг M^* равен четырём. Допустим, что $\{a, b, c\}$ — трёхэлементный цикл. Тогда он лежит в плоскости ранга два (так как M^* простой), которая должна быть пересечением гиперплоскостей, что противоречит предположению. Значит, все трёхэлементные подмножества независимы и каждое из них является базой единственной

(в силу предположения) гиперплоскости в M^*, которая сама поэтому представляет собой цикл. Отсюда каждое четырёхэлементное подмножество — либо цикл в M^*, если оно является четвёркой (т. е. гиперплоскостью в M^*), либо база в M^*, если не является. Поэтому каждое пятиэлементное подмножество зависимо (и само является циклом, если не содержит четвёрку).

Предположим, что имеется пересечение двух гиперплоскостей матроида M^* по трёхэлементному множеству, $|H_1^* \cap H_2^*| = 3$. Тогда для $C_i = E \setminus H_i^*$ $(i = 1, 2)$ имеем $|C_1 \oplus C_2| = 2$ и поэтому каждое n-элементное подмножество $(n+1)$-элементного множества $F = (C_1 \cup C_2)$ является циклом. Если F не замкнуто (т. е. не является плоскостью матроида M), то существует $a \in E \setminus F$ и цикл $C \in \mathcal{C}$, такой, что $C \setminus F = \{a\}$, но тогда, очевидно, и в множестве $\{a\} \cup F$ каждое n-элементное подмножество является циклом. Обозначим $\{a, b, c\} = E \setminus F$; в силу разделимости матроида M существует цикл, содержащий b, но не содержащий c, однако тогда и в множестве $\{a, b\} \cup F$ каждое n-элементное подмножество — цикл.

В силу связности матроида M найдётся цикл, соединяющий оставшийся элемент c с множеством $\{a, b\} \cup F$, откуда замыкание множества F совпадает с носителем E матроида M, причём каждое n-элементное его подмножество есть цикл, а это означает равномерность матроида M.

Для того чтобы матроид M не был равномерным, необходимо, чтобы F было его плоскостью. Однако в силу разделимости матроида M существует цикл C_a, такой, что $a \in C_a$, $b \notin C_a$. Ввиду замкнутости F необходимо $c \in C_a$; аналогично — найдётся цикл C'_a, такой, что $a \in C'_a$, $c \notin C'_a$, но $b \in C'_a$. Это означает, что F — гиперплоскость матроида M, и поэтому $E \setminus F$ есть цикл двойственного матроида M^*. Значит, цикл $\{a, b, c\}$ есть плоскость (как пересечение гиперплоскостей) в M^* ранга два (так как M^* простой) и поэтому гиперплоскости матроида M^* имеют ранг равный трём.

Пусть $\{b_1, b_2, b_3\}$ — база произвольной гиперплоскости H^*, не являющейся циклом матроида M. Поскольку $|H^*| = 4$, имеется единственный цикл, скажем, $\{b_1, b_2, h\}$, при $H^* = \{b_1, b_2, b_3, h\}$. Однако тогда для любого элемента $x \in E$, $x \notin H^*$ имеем: множества $\{b_1, b_2, x\}$, $\{b_1, b_3, x\}$, $\{b_2, b_3, x\}$ независимы, так как множество $\{b_1, b_2, b_3, x\}$ независимо и поэтому является базой матроида M^*. Зафиксируем $x = b_0$. Тогда для любого $y \notin \{b_0, b_1, b_2, b_3\} = B^*$ найдётся единственный цикл C^*, такой, что $C^* \setminus B^* = \{y\}$, и при $y \neq h$ необходимо $b_0 \in C^*$. Поскольку мощность гиперплоскости матроида M не может быть меньше n, если он не равномерный, то мощность её дополнения не может быть больше четырёх. Отсюда $|C^*| \leqslant 4$, в случае равенства C^* — гиперплоскость матроида M^* и поэтому его дополнение — цикл матроида M, являющийся также и его гиперплоскостью. ■

Утверждение 2. Если матроид связный и разделяющий, то двойственный ему матроид простой.

Доказательство. Пусть матроид $M = (E, \mathcal{C})$ связный, $|E| \geqslant 2$; тогда двойственный матроид $M^* = (E, \mathcal{H}^*)$ не имеет одноэлементных циклов. Действительно: если $\{e\}$ — цикл в M^* (т. е. коцикл в M), то $H = E \setminus \{e\} \neq \varnothing$ — гиперплоскость в M. Однако для любого $h \in H$ существует, ввиду связности, цикл C, содержащий и e, и h. При этом $|C| > 1$, $|C \setminus H| = |\{e\}| = 1$, откуда e принадлежит гиперплоскости H — противоречие.

Пусть матроид M разделяющий, $|E| \geqslant 3$. Тогда M^* не имеет двухэлементных циклов. Действительно: если $\{e, f\}$ — двухэлементный цикл в M^*, то $H = E \setminus \{e, f\} \neq \varnothing$ — гиперплоскость в M. В силу того, что M разделяющий, существует цикл, содержа-

щий e, но не содержащий f, и существует цикл, содержащий f, но не содержащий e. Эти циклы не могут быть одноэлементными, т. е. $\{e\}$ и $\{f\}$, по первой аксиоме циклов. Следовательно, существует цикл C, такой, что $|C| > 1$ и, без ограничения общности, $e \in C$, $f \notin C$. Но тогда $|C \setminus H| = |\{e\}| = 1$, откуда e принадлежит гиперплоскости H — противоречие. ■

Таким образом, вариантом однородного неравномерного матроида, которому не соответствует блок-схема, может быть только реализация случая с возможностью пересечения его когиперплоскостей по трёхэлементному множеству.

Утверждение 3. Однородный связный разделяющий матроид с мощностью циклов $n = 5$ является равномерным.

Доказательство. Пусть $M = (E, \mathcal{C})$; при $|E| = 5$ матроид не разделяющий, а при $|E| = 6$ он, очевидно, равномерный, так как тогда любые два различных цикла пересекаются по четырёхэлементному множеству и поэтому любое пятиэлементное подмножество их объединения, равного E, является циклом. Аналогично при $|E| = 7$: если $E = \{a, b\} \cup C$, где C — цикл, не содержащий ни a, ни b, то в силу разделимости существует цикл C_1, содержащий a, но не содержащий b, и тогда $|C_1 \cap C| = 4$, откуда во множестве $\{a\} \cup C$ каждое пятиэлементное подмножество есть цикл. В силу связности существующий цикл, содержащий $\{a, b\}$, пересекается с любым циклом, содержащим b, но не содержащим a, по четырёхэлементному множеству, откуда вытекает равномерность матроида M. Эти же рассуждения применимы при $|E| = 8$.

Пусть теперь $|E| \geqslant 9$, $e \in E$ — произвольный элемент матроида, $D \in \mathcal{C}$ — произвольный его цикл, не содержащий e. Тогда в силу связности этот цикл может быть представлен в виде $D = (C_1 \cup C_2) \setminus J_e(C_1, C_2)$, где C_1, C_2 — циклы, содержащие e, $J_e(C_1, C_2) = \bigcap\{C : e \in C \subset (C_1 \cup C_2)\}$.

Допустим, существует такой цикл D, что нет циклов, содержащих e и не пересекающихся с D по четырёхэлементному множеству. Отсюда $|C_1 \cap C_2| < 4$, и из $|D| = |C_1| = |C_2| = n = 5$ вытекает, что $|C_1 \cap C_2| > |J_e(C_1, C_2)| \geqslant 2$. Поэтому $|C_1 \cap C_2| = 3$, $|J_e(C_1, C_2)| = 2$.

Пусть $J_e(C_1, C_2) = \{e, f\}$, $C_1 \cap C_2 = \{e, f, g\}$. Тогда $D = (C_1 \oplus C_2) \cup \{g\}$ и существует такой цикл $C \subset (C_1 \cup C_2)$, что $\{e, f\} \subset C$, $g \notin C$. Отсюда следует, что $|(C_1 \oplus C_2) \cap C| = 3$. Однако тогда либо $|C_1 \cap C| = 4$, либо $|C_2 \cap C| = 4$.

Пусть, без ограничения общности, $|C_1 \cap C| = 4$. Тогда в шестиэлементном множестве $(C_1 \cup C)$ каждое пятиэлементное множество есть цикл, в том числе $C' = (D \cap (C_1 \cup \cup C)) \cup \{e\}$. Однако ясно, что $|C' \cap D| = 4$ вопреки предположению о D и e. Итак, от противного утверждение доказано. ■

Представленный подход может быть применён к решению более сложных задач обобщения связи блок-схем и однородных матроидов.

ЛИТЕРАТУРА

1. *Асанов М. О., Баранский В. А., Расин В. В.* Дискретная математика: графы, матроиды, алгоритмы. Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001. 288 с.
2. *Welsh D. J. A.* Matroid Theory. London: Academic Press, 1976.
3. *Парватов Н. Г.* Совершенные схемы разделения секрета // Прикладная дискретная математика. 2008. № 2(2). С. 50–57.
4. *Beimel A. and Livne N.* On matroids and non-ideal secret sharing // TCC 2006. LNCS. 2006. V. 3876. P. 482–501.

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. *Холл М.* Комбинаторика. М.: Мир, 1970.

УДК 511.48 DOI 10.17223/2226308X/13/3

АЛГОРИТМ ВЫЧИСЛЕНИЯ ЭЛЕМЕНТА ШТИКЕЛЬБЕРГЕРА ДЛЯ МНИМЫХ МУЛЬТИКВАДРАТИЧНЫХ ПОЛЕЙ

Д. О. Олефиренко, Е. А. Киршанова, Е. С. Малыгина, С. А. Новоселов

Представлен алгоритм вычисления идеала Штикельбергера для мультиквадратичного поля $K = \mathbb{Q}(\sqrt{d_1}, \sqrt{d_2}, \ldots, \sqrt{d_n})$, где $d_i \equiv 1 \pmod 4$, $i = 1, \ldots, n$, и d_i попарно взаимно просты. Мы алгоритмизируем идеи, описанные в работе Р. Кучеры 1996 г., доказываем корректность полученных алгоритмов и анализируем их сложность. Для $2^n = [K : \mathbb{Q}]$ алгоритм работает за время $\widetilde{\mathcal{O}}(2^n)$. Полученный результат полезен для решения криптоаналитических задач поиска короткого вектора в идеалах мультиквадратичных полей.

Ключевые слова: *мультиквадратичные поля, идеал Штикельбергера, элемент Штикельбергера, задача поиска короткого вектора.*

Введение

Получение идеала Штикельбергера в явном виде является важной алгоритмической задачей в вычислительной теории чисел, в теории групп классов и, с недавних пор, в криптоанализе. Для числового поля K идеал Штикельбергера I — идеал групповой алгебры $\mathbb{Z}[G_K]$, где $G_K = \mathrm{Gal}(K/\mathbb{Q})$ — группа Галуа поля K. Полезное свойство I заключается в том, что под действием элементов I на Cl_K — группу классов идеалов K — любой класс становится тривиальным (иначе говоря, J^σ — главный идеал для любого $\sigma \in I$ и любого идеала J кольца целых $\mathcal{O}_K$ числового поля K).

Для кругового поля $K = \mathbb{Q}(\zeta_n)$ идеал Штикельбергера, рассматриваемый как решётка в $\mathbb{Z}^n$ с помощью вложения $\mathbb{Z}[G_K] \hookrightarrow \mathbb{Z}^n$, обладает «хорошим» базисом. Явный вид этого базиса описан, например, в [1]. Это свойство идеала Штикельбергера в сочетании с 1) «обнуляющим» действием элементов идеала на целые идеалы $\mathbb{Z}[\zeta_n]$ и 2) существованием (относительно) быстрого алгоритма нахождения короткого вектора в *главных* идеалах $\mathbb{Z}[\zeta_n]$ позволило получить алгоритм нахождения короткого вектора в идеалах кольца целых круговых полей [1]. В современном криптоанализе нахождение короткого вектора в решётках является основополагающей задачей.

Именно приложение идеала Штикельбергера в криптоанализе является нашей главной мотивацией для его изучения. Ввиду большой группы Галуа интересными полями являются мультиквадратичные. Недавние работы по эффективному вычислению короткого вектора в мультиквадратичных полях [2] и по вычислению группы классов [3] подводят к вопросу нахождения коротких векторов в *произвольных* идеалах. Следуя примеру круговых полей, логично обратить внимание на структуру идеала Штикельбергера для мультиквадратичных расширений.

В работе предлагается алгоритм вычисления идеала Штикельбергера для мультиквадратичного поля $K = \mathbb{Q}(\sqrt{d_1}, \sqrt{d_2}, \ldots, \sqrt{d_n})$, где $d_1 \equiv d_2 \equiv \ldots \equiv d_n \equiv 1 \pmod 4$ и d_i попарно взаимно просты. Алгоритм имеет сложность $\widetilde{\mathcal{O}}(2^n)$, а так как $[K : \mathbb{Q}] = 2^n$, то даже для криптографически значимых степеней он эффективен. В основе алгорит-

ма лежит работа Р. Кучеры [4]. Доказательства всех утверждений статьи можно найти в полной версии работы[1].

1. Предварительные сведения

Исходя из условий, наложенных на d_i, справедливо вложение $K \hookrightarrow \mathbb{Q}(\zeta_{d_1 \cdot \ldots \cdot d_n})$. Кроме того, $K \cap \mathbb{Q}(\zeta_{d_1 \cdot \ldots \cdot d_\ell}) = \mathbb{Q}(\sqrt{d_1}, \sqrt{d_2}, \ldots, \sqrt{d_\ell})$, где $\ell \leqslant n$, что доказано в лемме 1.

Рассмотрим числовые поля $\mathbb{Q} \subseteq K \subseteq L$ и обозначим их соответствующие группы Галуа $G_L = \mathrm{Gal}(L/\mathbb{Q})$ и $G_K = \mathrm{Gal}(K/\mathbb{Q})$; $\mathbb{Q}[G_L] = \{\sum a_i\sigma_i : a_i \in \mathbb{Q}, \sigma_i \in G_L\}$ и $\mathbb{Q}[G_K] = \{\sum a_i\sigma_i : a_i \in \mathbb{Q}, \sigma_i \in G_K\}$ — группы, конечно порождённые элементами G_L (соответственно элементами G_K) над $\mathbb{Q}$. Важными понятиями при вычислении элементов Штикельбергера являются отображения res и cor. Определим эти отображения, согласно [4], для расширения L/K:

$$\mathsf{res}_{L/K} : \mathbb{Q}[G_L] \to \mathbb{Q}[G_K], \quad \mathsf{res}_{L/K}\Big(\sum_{\sigma \in G_L} a_\sigma \sigma\Big) = \sum_{\sigma \in G_L} a_\sigma(\sigma|_K),$$
$$\mathsf{cor}_{L/K} : \mathbb{Q}[G_K] \to \mathbb{Q}[G_L] \quad \mathsf{cor}_{L/K}\Big(\sum_{\sigma \in G_K} a_\sigma \sigma\Big) = \sum_{\sigma \in G_K} a_{\sigma|_K} \sigma,$$

где $\sigma|_K$ означает сужение автоморфизма $\sigma \in G_L$ на поле K; a_σ, $a_{\sigma|_K}$ — коэффициенты, соответствующие автоморфизмам σ, $\sigma|_K$.

Дробную часть числа обозначим $\langle\cdot\rangle$, т. е. $0 < \langle\cdot\rangle < 1$; наибольший общий делитель элементов $a, b \in \mathbb{Z}$ — через (a, b); символ Лежандра этих же элементов — $\left(\frac{a}{b}\right)$. Для произвольного множества A его мощность обозначим $\#A$. Дадим классические определения элемента и идеала Штикельбергера, согласно [5, с. 189].

Определение 1. Для любых $n \in \mathbb{N}$ и $\alpha \in \mathbb{Z}$ и кругового поля $\mathbb{Q}(\zeta_n)$ определим

$$\theta_n(\alpha) = \sum_{(a,n)=1} \left\langle -\frac{\alpha a}{n} \right\rangle \sigma_a^{-1},$$

где $0 < a \leqslant n$ и $\sigma_a \in G_{\mathbb{Q}(\zeta_n)} = \mathrm{Gal}(\mathbb{Q}(\zeta_n)/\mathbb{Q})$.

Определение 2. Для любых $n \in \mathbb{N}$ и $\alpha \in \mathbb{Z}$ определим

$$\theta_n'(\alpha) = \left(\mathsf{cor}_{K/K\cap\mathbb{Q}(\zeta_n)} \circ \mathsf{res}_{\mathbb{Q}(\zeta_n)/K\cap\mathbb{Q}(\zeta_n)}\right)(\theta_n(\alpha))$$

— элемент Штикельбергера, где K и $\mathbb{Q}(\zeta_n)$ — соответственно числовое и круговое поля.

Определение 3. Идеалом Штикельбергера поля K называется идеал вида

$$I = \{\theta_n'(\alpha) | \alpha, n \in \mathbb{Z}, n \geqslant 1\} \cap \mathbb{Z}[G_K].$$

Теперь дадим определение квадратичным гауссовым суммам, а также покажем, как они взаимосвязаны с автоморфизмами круговых полей.

Определение 4. Пусть $m, k \in \mathbb{Z}$, $k > 0$. Квадратичная гауссова сумма определяется как $g(m, k) = \sum\limits_{b=0}^{k-1} \mathrm{e}^{2\pi i m b^2/k}$.

Следующая теорема позволяет выражать квадратные корни, рассматриваемые как элементы мультиквадратичного поля, через квадратичные гауссовы суммы.

[1] Представлена на страницах авторов `https://crypto-kantiana.com/`.

Теорема 1 [6, 1.5.2, с. 26]. Пусть $(m,k)=1$, $k>0$ и k нечётное. Тогда

$$g(m,k)=\left(\frac{m}{k}\right)g(1,k)=\begin{cases}\left(\dfrac{m}{k}\right)\sqrt{k}, & k\equiv 1 \pmod 4,\\ \left(\dfrac{m}{k}\right)i\sqrt{k}, & k\equiv 3 \pmod 4.\end{cases}$$

Если $-k\equiv 1 \pmod 4$, то $k\equiv 3 \pmod 4$. Тогда $\sqrt{-k}=g(1,k)$ по теореме 1. Рассмотрим отображение $\mathsf{res}_{\mathbb{Q}(\zeta_n)/K\cap\mathbb{Q}(\zeta_n)}$. Прокомментируем, как происходит сужение автоморфизмов поля $\mathbb{Q}(\zeta_n)$ на поле $K\cap\mathbb{Q}(\zeta_n)$ и что есть пересечение $K\cap\mathbb{Q}(\zeta_n)$.

Ответим на первый вопрос, определив сужение кругового поля $\mathbb{Q}(\zeta_n)$ на некоторое числовое поле. Рассмотрим общий случай, когда $n=pq$, где $p,q>0$ — взаимно простые. Тогда автоморфизм σ_a поля $\mathbb{Q}(\zeta_{pq})$ можно связать с действием автоморфизмов полей $\mathbb{Q}(\zeta_p)$ и $\mathbb{Q}(\zeta_q)$ на элементы $\sqrt{-p}$ и $\sqrt{-q}$ следующим образом:

$$\sigma_a(\zeta_{pq})=g(a,pq)=\left(\frac{aq}{p}\right)g(1,p)\left(\frac{ap}{q}\right)g(1,q)=\sigma_{aq}(\sqrt{-p})\sigma_{ap}(\sqrt{-q}).$$

Здесь индекс aq в случае $\sigma_{aq}(\sqrt{-p})$ рассматривается по модулю p, индекс ap в случае $\sigma_{ap}(\sqrt{-q})$ — по модулю q. Более детально все вычисления представлены в п. 2. Ответ на второй вопрос даёт следующая

Лемма 1. Пусть $K=\mathbb{Q}(\sqrt{d_1},\sqrt{d_2},\ldots,\sqrt{d_n})$ — мультиквадратичное поле, где $d_1\equiv$ $\equiv d_2\equiv\ldots\equiv d_n\equiv 1 \pmod 4$ и все d_i попарно взаимно просты для $i=1,\ldots,n$; $\mathbb{Q}(\zeta_{d_1\cdot\ldots\cdot d_\ell})$ — круговое поле, где $\zeta_{d_1\cdot\ldots\cdot d_\ell}$ — корень степени $d_1\cdot\ldots\cdot d_\ell$ из единицы и $\ell\leqslant n$. Тогда $K\cap\mathbb{Q}(\zeta_{d_1\cdot\ldots\cdot d_\ell})=\mathbb{Q}(\sqrt{d_1},\sqrt{d_2},\ldots,\sqrt{d_\ell})$.

Для упрощения дальнейших вычислений дадим альтернативное определение идеалу Штикельбергера; его эквивалентность исходному определению, обеспечивающая корректность работы алгоритма, представлена в полной версии статьи на `https://crypto-kantiana.com/`. Пусть f — кондуктор поля K, тогда $K\cap\mathbb{Q}(\zeta_n)=K\cap\mathbb{Q}(\zeta_{(f,n)})$ для $n\in\mathbb{N}$. Определение кондуктора числового поля можно посмотреть в [7].

Определение 5. Идеалом Штикельбергера поля K с кондуктором f называется идеал вида $I=I'\cap\mathbb{Z}[G_K]$, где

$$I'=\{\sigma\cdot\theta_n'(-1):n|f,\sigma\in G_K\}\bigcup\left\{\frac{1}{2}N_K\right\}.$$

2. Алгоритм

Рассмотрим алгоритм вычисления идеала Штикельбергера для мнимых мультиквадратичных полей в соответствии с описанной в п. 1 теорией.

Вычисление действия отображения res не тривиально в общем случае, поэтому рассмотрим его более детально. Применим квадратичные гауссовы суммы для вычисления $\mathsf{res}_{\mathbb{Q}(\zeta_{d_1\cdot\ldots\cdot d_n})/K\cap\mathbb{Q}(\zeta_{d_1\cdot\ldots\cdot d_n})}\theta_{d_1\cdot\ldots\cdot d_n}(-1)$, где

$$\theta_{d_1\cdot\ldots\cdot d_n}(-1)=\sum_{(a,d_1\cdot\ldots\cdot d_n)=1}\left\langle\frac{a}{d_1\cdot\ldots\cdot d_n}\right\rangle\sigma_a^{-1}. \tag{1}$$

Здесь $\sigma_a\in G_{\mathbb{Q}(\zeta_{d_1\cdot\ldots\cdot d_n})}$ действуют лишь на элемент $\sqrt{d_1\cdot\ldots\cdot d_n}$. По формуле (1) каждый такой автоморфизм сводится к произведению автоморфизмов полей $\mathbb{Q}(\zeta_{d_1})$,

$\ldots$, $\mathbb{Q}(\zeta_{d_n})$. Рассмотрим каждое слагаемое $\left\langle \dfrac{a}{d_1 \cdot \ldots \cdot d_n} \right\rangle \sigma_a^{-1}$. Введём обозначение $\pi_i = a \prod\limits_{j \neq i} d_j$, тогда

$$\left\langle \frac{a}{d_1 \cdot \ldots \cdot d_n} \right\rangle \sigma_a^{-1}(\zeta_{d_1 \cdot \ldots \cdot d_n}) = \sigma_a(\sqrt{d_1 \cdot \ldots \cdot d_n}) =$$
$$= \left\langle \frac{a}{d_1 \cdot \ldots \cdot d_n} \right\rangle \sigma_{\frac{1}{\pi_1 \bmod d_1} \bmod d_1}(\zeta(d_1)) \cdot \ldots \cdot \sigma_{\frac{1}{\pi_n \bmod d_n} \bmod d_n}(\zeta(d_n)).$$

Если $\frac{1}{\pi_1 \bmod d_1} \bmod d_1$ — квадратичный вычет по модулю d_1, то заменяем $\sigma_{\frac{1}{\pi_1 \bmod d_1} \bmod d_1}(\zeta(d_1))$ на id_1, где $id_1 : \sqrt{d_1} \to \sqrt{d_1}$; в противном случае — на σ_1, где $\sigma_1 : \sqrt{d_1} \to -\sqrt{d_1}$. Аналогично рассуждаем для остальных множителей.

Полученные композиции автоморфизмов переобозначим следующим образом:

$$\begin{gathered} id_1 \cdot id_2 \cdot \ldots \cdot id_n = id : \sqrt{d_1} + \ldots + \sqrt{d_n} \to \sqrt{d_1} + \sqrt{d_2} + \ldots + \sqrt{d_n}, \\ \ldots \\ \sigma_1 \cdot \ldots \cdot \sigma_{n-1} \cdot \sigma_n = \tau_m : \sqrt{d_1} + \ldots + \sqrt{d_{n-1}} + \sqrt{d_n} \to -\sqrt{d_1} - \ldots - \sqrt{d_n}. \end{gathered} \tag{2}$$

Очевидно, что общее количество получившихся композиций автоморфизмов равно 2^n. Поскольку первый автоморфизм обозначен id, то $m = 2^n - 1$. Процедура вычисления res представлена в алгоритме 1.

Алгоритм 1. Вычисление $\mathsf{res}(\theta_n(-1))$

Вход: $K = \mathbb{Q}(\sqrt{d'_1}, \sqrt{d'_2}, \ldots, \sqrt{d'_k})$.
Выход: $\mathsf{res}(\theta_n(-1))$.

1: $f := \prod\limits_{j=1}^{k} d'_j$. // f — *кондуктор* K
2: **Для** $a \in \mathbb{Z}_f^*$
3: **Для** $j = 1, \ldots, k$
4: $t := \dfrac{a \cdot d'_1 \cdot \ldots \cdot d'_k}{d'_j} \bmod d'_j$;
5: $index := \dfrac{1}{t} \bmod d'_j$.
6: **Если** $\left(\dfrac{index}{d'_j}\right) = 1$, **то**
7: $\sigma_a^{-1} := \sigma_a^{-1} \cdot id_j$, // id_j — *тождественный в* $\mathbb{Q}(\sqrt{d'_j})$
8: **иначе Если** $\left(\dfrac{index}{d'_j}\right) = -1$, **то**
9: $\sigma_a^{-1} := \sigma_a^{-1} \cdot \sigma_j$; // σ_j — *сопряжение в* $\mathbb{Q}(\sqrt{d'_j})$
10: $\sigma_a^{-1} := \dfrac{a}{f} \cdot \sigma_a^{-1}$;
11: $\theta := \sum\limits_{a \in \mathbb{Z}_f^*} \sigma_a^{-1}$;
12: res: заменить получившиеся комбинации автоморфизмов в каждом слагаемом θ на соответствующие τ_i в соответствии с формулами (2).
13: **Вернуть** res.

Вычисление cor**.** Автоморфизмы, полученные после вычисления res, являются автоморфизмами поля $K \cap \mathbb{Q}(\zeta_{d_1 \cdot \ldots \cdot d_n})$. Вычисление действия отображения cor представляет собой переход от этих автоморфизмов к автоморфизмам поля K. Обозначим автоморфизмы поля K следующим образом: сопоставим действие автоморфизма ρ_i с бинарным вектором из $\mathbb{Z}_2^n$, причём если i-я координата вектора (считая слева направо) есть 1, то $\rho_i : \sqrt{d_i} \to -\sqrt{d_i}$ (например, $\rho_1 : \sqrt{d_1} + \ldots + \sqrt{d_n} \to \sqrt{d_1} + \ldots - \sqrt{d_n}$).

Если $K \cap \mathbb{Q}(\zeta_{d_1 \cdot \ldots \cdot d_n}) = K$, то отображение cor действует тождественно (полученные автоморфизмы τ_i совпадают с ρ_i). Такой случай возникает при вычислении $\theta'_{d_1 \cdot \ldots \cdot d_n}(-1)$. А как быть, если мы вычисляем, например, элемент Штикельбергера вида $\theta'_{d_1 \cdot \ldots \cdot d_\ell}(-1)$, где $\ell < n$? Рассмотрим случай, когда $K \cap \mathbb{Q}(\zeta_{d_1 \cdot \ldots \cdot d_\ell}) = \mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_\ell})$. Результатом действия отображения res в этом случае является

$$a_1 \cdot id_l + a_2 \cdot \tau_1 + \ldots + a_{2^l-1} \cdot \tau_{2^l-2} + a_{2^l} \cdot \tau_{2^l-1},$$

где id_l, τ_i — автоморфизмы поля $\mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_\ell})$, $i = 1, \ldots, 2^l - 1$. Нумерация автоморфизмов τ_i аналогична нумерации автоморфизмов ρ_i.

Далее переходим от перечисленных автоморфизмов поля $\mathbb{Q}(\sqrt{d_1}, \ldots, \sqrt{d_\ell})$ к автоморфизмам ρ_i поля K. Если ρ_i относительно элемента $\sqrt{d_1} + \ldots + \sqrt{d_l}$ действует как id_l, то все такие автоморфизмы ρ_i участвуют в записи элемента Штикельбергера с коэффициентом a_1. Аналогично, если ρ_i относительно $\sqrt{d_1} + \ldots + \sqrt{d_l}$ действует как τ_1, то все такие автоморфизмы ρ_i участвуют в записи элемента Штикельбергера с коэффициентом a_2. Применяя такой подход для всех остальных случаев, получаем

$$\theta'_{d_1 \cdot \ldots \cdot d_\ell}(-1) = a_1 \cdot id + a_1 \cdot \rho_1 + \ldots + a_{2^l} \cdot \rho_{m-1} + a_{2^l} \cdot \rho_m.$$

Таким образом, в общем случае элемент Штикельбергера примет вид

$$\theta'_n(-1) = c_0 \cdot id + c_1 \cdot \rho_1 + \ldots + c_{m-1} \cdot \rho_{m-1} + c_m \cdot \rho_m,$$

где $c_i \in \mathbb{Z}$ для $i = 0, \ldots, m$ и $m = 2^n - 1$.

Очевидно, что общее количество элементов Штикельбергера в поле K равно $2^n - 1$ (по количеству всех возможных подполей). Таким образом, необходимо умножить 2^n автоморфизов на каждый из $2^n - 1$ элементов Штикельбергера.

Количество различных комбинаций зависит от количества различных коэффициентов в элементе Штикельбергера. Будем записывать все различные комбинации в множество I'; общее количество различных элементов для мультиквадратичного поля будет равно $\#I'$. В результате получим следующий результат:

$$I = s_1 \cdot \theta_1 + \ldots + s_{\#I'} \cdot \theta_{\#I'} = \sum_{i=1}^{\#I'} s_i \cdot \theta_i, \quad s_i \in \mathbb{Z}.$$

Описанные действия представлены в алгоритме 2.

Алгоритм 2. Вычисление идеала Штикельбергера

Вход: $K = \mathbb{Q}(\sqrt{d_1}, \sqrt{d_2}, \ldots, \sqrt{d_n})$.
Выход: $I = I' \cap \mathbb{Z}[Gal(K/\mathbb{Q})]$.

1: Построить массив A, состоящий из всех подполей K.
2: **Для** $i = 1, \ldots, 2^n - 1$:
3: $\quad \mathsf{res}_i := \mathsf{res}_{\mathbb{Q}(\zeta_{d'_1 \cdot \ldots \cdot d'_k})/\mathbb{Q}(\sqrt{d'_1}, \sqrt{d'_2}, \ldots, \sqrt{d'_k})} \theta_i$; // алгоритм 1, вход: $A[i]$
4: $\quad \theta'_i := \mathsf{cor}_{K/\mathbb{Q}(\sqrt{d'_1}, \sqrt{d'_2}, \ldots, \sqrt{d'_k})} \mathsf{res}_i$;
5: $I' - \varnothing$;
6: **Для** $i = 1, \ldots, 2^n - 1$
7: $\quad$ **Для** $j = 1, \ldots, 2^n$
8: $\quad\quad t := \rho_j \cdot \theta'_i$; $I' := A \cup t$;
9: $I := \prod_{i=1}^{\#I'} s_i \cdot I'_i$.
10: **Вернуть** I.

Лемма 2. Пусть $d = \max\limits_i d_i$. Тогда вычислительная сложность алгоритма 2 равна

$$\mathcal{O}(e^{n \log n} \cdot 2^{2n} \cdot n^4 \cdot \log^3 d \cdot \log^3 n).$$

ЛИТЕРАТУРА

1. *Cramer R., Ducas L., and Wesolowski B.* Short Stickelberger class relations and application to ideal-SVP // Advances in Cryptology — Eucrocrypt 2017. Springer, 2017. P. 324–348.
2. *Bauch J., Bernstein D. J., de Valence H., et al.* Short generators without quantum computers: The case of multiquadratics // Advances in Cryptology — EUROCRYPT 2017. Springer, 2017. P. 27–59.
3. *Biasse J.-F. and Vredendaal C.* Fast multiquadratic S-unit computation and application to the calculation of class groups // Open Book Series. 2019. V. 2. P. 103–118.
4. *Kucera R.* On the Stickelberger ideal and circular units of a compositum of quadratic fields // J. Number Theory. 1996. V. 56. No. 1. P. 139–166.
5. *Sinnott W.* On the Stickelberger ideal and the circular units of an Abelian field // Invent. Math. 1980. V. 62. P. 181–234.
6. *Berndt B. C., Evans R. J., and Williams K. S.* Topics in Commutative Ring Theory. N.Y.: Wiley, 1998.
7. *Cohen H. and Stevenhagen P.* Computational Class Field Theory. 2008. `https://arxiv.org/pdf/0802.3843.pdf`

Секция 2

ДИСКРЕТНЫЕ ФУНКЦИИ

УДК 519.7 DOI 10.17223/2226308X/13/4

О ПРОДОЛЖЕНИИ ДО БЕНТ-ФУНКЦИЙ И ОЦЕНКЕ СВЕРХУ ИХ ЧИСЛА

С. В. Агиевич

Булева бент-функция f от n переменных является продолжением булевой функции g от $k < n$ переменных, если g является сужением f на фиксированную аффинную плоскость размерности k. Доказывается, что продолжение всегда существует, если $k \leqslant n/2$. Получена оценка сверху для числа продолжений. Оценка усиливается для случая $k = n - 1$, когда g является почти-бент-функцией. В результате мы улучшаем известные оценки сверху для числа бент-функций.

Ключевые слова: *бент-функция, число бент-функций, почти-бент-функция, аффинная плоскость.*

Булева функция $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$ от чётного числа переменных n называется *бент-функцией*, если $|\hat{f}(\mathbf{u})| = 2^{n/2}$ для всех $\mathbf{u} \in \mathbb{F}_2^n$. Здесь $\hat{f}$ — спектр Уолша — Адамара функции f:

$$\hat{f}(\mathbf{u}) = \sum_{x \in \mathbb{F}_2^n} \chi(f(\mathbf{x}) + \mathbf{x} \cdot \mathbf{u}).$$

Символ χ под знаком суммы — это нетривиальный аддитивный характер $\mathbb{F}_2$: $\chi(a) = (-1)^a$, точка обозначает скалярное произведение векторов.

Пусть $\mathcal{B}_n$ — множество всех бент-функций от n переменных. Точное значение $|\mathcal{B}_n|$ неизвестно уже для $n = 10$, более того, адекватное оценивание $|\mathcal{B}_n|$ как сверху, так и снизу остаётся трудной задачей (см. обсуждение в [1]). В настоящей работе нас интересуют оценки сверху.

Обозначим $B(d, n) = 2^{\binom{n}{0}+\binom{n}{1}+\ldots+\binom{n}{d}}$ и напомним, что булева функция f однозначно представляется многочленом фактор-кольца $\mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 - x_1, \ldots, x_n^2 - x_n)$. Пусть $\deg f$ — степень многочлена.

Наивная оценка сверху (так она названа в работе [2]) для $|\mathcal{B}_n|$ основана на том, что если $f \in \mathcal{B}_n$ и $n \geqslant 4$, то $\deg f \leqslant n/2$. Оценка имеет следующий вид:

$$|\mathcal{B}_n| \leqslant B(n/2, n) = 2^{2^{n-1}+\binom{n}{n/2}/2} \approx 2^{2^{n-1}+2^n/\sqrt{2\pi n}}.$$

Оценка может быть немного усилена: следует учесть условие $2 \leqslant \deg f$ и вычесть из правой части 2^{n+1} — число аффинных функций.

В [2] К. Карле и А. Клаппер нашли более серьёзное усиление:

$$|\mathcal{B}_n| \leqslant \frac{B(n/2, n)}{2^{2^{n/2}-n/2-1}}(1 + \varepsilon_n) + B(n/2 - 1, n), \quad \varepsilon_n = \frac{1}{2^{\binom{n-1}{n/2-1}-2}},$$

справедливое для $n \geqslant 6$. Эта оценка считается лучшей на сегодняшний день. В [2], кроме ограничения на $\deg f$, учитывается также спектральное строение бент-функций. Мы улучшаем оценку Карле — Клаппера.

Теорема 1. При чётном $n \geqslant 6$ справедлива оценка

$$|\mathcal{B}_n| \leqslant c_n\, 2^{2^{n-2}-n/2+5/2} \left(\frac{B(n/2, n-1) - B(n/2-1, n-1)}{2^{2^{n/2}-n/2-1}} + B(n/2-1, n-1) \right),$$

в которой $c_n = \exp(-1/2 + 23/(18 \cdot 2^{n-2}))/\sqrt{\pi}$, причём $c_n \leqslant c_6 \approx 0{,}3706$.

Различные оценки сверху для $|\mathcal{B}_n|$ при малых n сведены в табл. 1. Точные значения $|\mathcal{B}_6| = 5425430528$ и $|\mathcal{B}_8| = 99270589265934370305785861242880$ найдены в работах [3] и [4] соответственно.

Таблица 1

n	$\lvert\mathcal{B}_n\rvert$	Оценки сверху для $\lvert\mathcal{B}_n\rvert$		
		Наивная	[2]	Настоящая работа
2	8			
4	896	2032		
6	$\approx 2^{32,3}$	2^{42}	2^{38}	2^{36}
8	$\approx 2^{106,3}$	2^{163}	2^{152}	2^{149}
10	?	2^{638}	2^{612}	2^{608}
12	?	2^{2510}	2^{2453}	2^{2448}

Метод оценивания основан на подсчёте числа продолжений булевой функции g от $k < n$ переменных до бент-функций от n переменных. Функция $f \in \mathcal{B}_n$ является *продолжением* g, если

$$g(y_1, \ldots, y_k) = f(\underbrace{0, \ldots, 0}_{n-k}, y_1, \ldots, y_k).$$

Другими словами, f — продолжение g, если g является сужением f на аффинную плоскость $E = \{(0, \ldots, 0, y_1, \ldots, y_k)\}$. Выбор E здесь не имеет принципиального значения, можно зафиксировать любую другую плоскость размерности k.

Пусть $\mathcal{B}_n(g)$ — множество всех функций $f \in \mathcal{B}_n$, которые являются продолжениями g. При доказательстве теоремы 1 мы рассматривали функции g от $n-1$ переменных, для которых $\mathcal{B}_n(g) \neq \varnothing$. Если g является подходящей, то значения $\hat{g}$ принадлежат множеству $\{0, \pm 2^{n/2}\}$ (и тогда g называется *почти-бент-функцией*) и, кроме этого, выполняется условие $\deg g \leqslant n/2$. Для оценки числа подходящих функций g мы применили результаты работы [2].

Для оценки $|\mathcal{B}_n(g)|$ использована следующая лемма, доказанная с помощью техники работы [5].

Лемма 1. Пусть N — чётное, S_N — сумма N независимых случайных величин с равномерным распределением на $\{-1, 1\}$. Для $s = 0, \pm 2, \ldots, \pm N$ справедлива следующая оценка:

$$\mathsf{P}[S_N = s] = \binom{N}{(N+s)/2} 2^{-N} \leqslant \sqrt{\frac{2}{\pi N}} \exp\left(-\frac{s^2}{2N} + \frac{23}{18N} \right).$$

Лемма 1 имеет и самостоятельное значение. С её помощью можно оценивать (сверху) биномиальные коэффициенты, контролировать точность аппроксимации в локальной теореме Муавра — Лапласа. В нашем контексте лемма позволяет оценить вероятность того, что спектральный коэффициент случайной булевой функции принимает заданное значение.

Оценку леммы 1 можно несколько улучшить, это улучшение потребуется в теореме 2. Речь идёт об оценке вида

$$\mathsf{P}[S_N = s] \leqslant 2^{-\alpha_N s^2 - \beta_N},$$

где α_N и β_N настраиваются так, чтобы величина $\gamma_N = \alpha_N + \beta_N/N$ была максимальной.

При малых N оптимальные тройки $(\alpha_N, \beta_N, \gamma_N)$ можно определить, решая задачи линейного программирования. Решения представлены в табл. 2.

Таблица 2

N	α_N	β_N	γ_N
2	1/2	1	3/4
4	1/6	4/3	1/2
8	1/12	$14/3 - \log_2 7$	$2/3 - \frac{1}{8}\log_2 7 \approx 0{,}3157$

В общем случае из леммы 1 следует, что

$$\gamma_N \geqslant \frac{\log_2 e + \log_2 \pi + \log_2 N - 1}{2N} - \frac{23 \log_2 e}{18N^2}.$$

С точки зрения теории бент-прямоугольников [6] величина $|\mathcal{B}_n(g)|$ — это число прямоугольников размерности $(n-k) \times k$, у которых первая строка фиксирована — она заполнена значениями $\hat{g}$. Учитывая ограничения на строки и столбцы бент-прямоугольника (точнее, тождества Парсеваля для них), получаем следующий результат.

Теорема 2. Для булевой функции g от $k < n$ переменных справедлива оценка

$$\log_2 |\mathcal{B}_n(g)| \leqslant 2^n (1 - \gamma_{2^{n-k}}).$$

Отметим, что оценка теоремы 2 с $k = n-1$ несколько усиливается при доказательстве теоремы 1.

Начиная с $k = n/2 + 1$, появляются функции g, которые нельзя продолжить до бент-функций. В этом можно убедиться, анализируя ограничения на столбцы бент-прямоугольника. Впрочем, оказывается, что

Теорема 3. При чётном n любая булева функция от $k \leqslant n/2$ переменных может быть продолжена до бент-функции от n переменных.

Теорему достаточно доказать для $k = n/2$. В этом случае с помощью биаффинной конструкции, предложенной в [7], можно построить бент-квадрат размерности $k \times k$, все строки и столбцы которого являются аффинными перестановками значений $\hat{g}$. Легко добиться, чтобы первая строка квадрата в точности совпадала с $\hat{g}$.

ЛИТЕРАТУРА

1. *Tokareva N.* Bent Functions: Results and Applications to Cryptography. London, UK; San Diego, CA, USA: Academic Press, 2015.
2. *Carlet C. and Klapper A.* Upper bounds on the numbers of resilient functions and of bent functions // Proc. 23rd Symp. Inform. Theory. Louvain-La-Neuve, Belgium. 2002. P. 307–314.
3. *Preneel B., Van Leekwijck W., Van Linden L., et al.* Propagation characteristics of Boolean functions // EUROCRYPT'90. LNCS. 1991. V. 473. P. 161–173.
4. *Langevin P. and Leander G.* Counting all bent functions in dimension eight 99270589265934370305785861242880 // Des. Codes Cryptogr. 2011. V. 59. P. 193–205.

5. *Szabados T.* A Simple Wide Range Approximation of Symmetric Binomial Distributions. Preprint arXiv:1612.01112 [math.PR]. 2016.
6. *Agievich S.* On the representation of bent functions by bent rectangles // Probabilistic Methods in Discrete Mathematics: Fifth Intern. Conf. (Petrozavodsk, Russia, June 1–6, 2000). Utrecht, Boston: VSP, 2002. P. 121–135.
7. *Agievich S.* Bent rectangles // Proc. NATO Advanced Study Institute on Boolean Functions in Cryptology and Information Security (Moscow, September 8–18, 2007). Amsterdam: IOS Press, 2008. P. 3–22.

УДК 519.7 DOI 10.17223/2226308X/13/5

О МЕТРИЧЕСКИХ СВОЙСТВАХ МНОЖЕСТВА САМОДУАЛЬНЫХ БЕНТ-ФУНКЦИЙ[1]

А. В. Куценко

Приводится обзор известных метрических свойств множества самодуальных бент-функций. Бент-функция называется самодуальной, если она совпадает со своей дуальной бент-функцией, и анти-самодуальной, если совпадает с отрицанием своей дуальной. Приводится полный спектр расстояний Хэмминга между самодуальными бент-функциями из класса Мэйорана — МакФарланда. Даются результаты, касающиеся характеризации булевых функций, находящихся на максимально возможном удалении от множества самодуальных бент-функций. Описаны группы автоморфизмов множеств самодуальных и анти-самодуальных бент-функций от n переменных, автоморфизмы множества булевых функций от n переменных, которые меняют местами множества самодуальных и анти-самодуальных бент-функций, изометричные отображения, сохраняющие неизменным отношение Рэлея каждой булевой функции от n переменных. Даётся характеризация всех изометричных отображений, сохраняющих максимальную нелинейность и расстояние Хэмминга между каждой бент-функций и дуальной к ней.

Ключевые слова: *булева функция, самодуальная бент-функция, расстояние Хэмминга, изометричное отображение, метрическая регулярность, группа автоморфизмов, отношение Рэлея.*

Через $\mathbb{F}_2^n$ обозначим линейное пространство всех двоичных векторов длины n над полем $\mathbb{F}_2$. *Булевой функцией* от n переменных называется отображение вида $\mathbb{F}_2^n \to \mathbb{F}_2$. Множество всех булевых функций от n переменных обозначается через $\mathcal{F}_n$. Для каждой пары $x, y \in \mathbb{F}_2^n$ через $\langle x, y\rangle$ обозначим скалярное произведение $\bigoplus\limits_{i=1}^{n} x_i y_i$. *Весом Хэмминга* $\mathrm{wt}(x)$ вектора $x \in \mathbb{F}_2^n$ называется число его ненулевых координат. *Расстояние Хэмминга* между булевыми функциями f, g от n переменных — число двоичных векторов длины n, на которых эти функции принимают различные значения, обозначается $\mathrm{dist}(f, g)$. *Преобразованием Уолша — Адамара* булевой функции f от n переменных называется целочисленная функция $W_f : \mathbb{F}_2^n \to \mathbb{Z}$, заданная равенством

$$W_f(y) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus \langle x, y\rangle}, \quad y \in \mathbb{F}_2^n.$$

[1]Работа выполнена в рамках государственного задания ИМ СО РАН (проект №0314-2019-0017) при поддержке РФФИ (проекты № 18-07-01394, 20-31-70043) и Лаборатории криптографии JetBrains Research.

Булева функция f от чётного числа переменных n называется *бент-функцией*, если $|W_f(y)| = 2^{n/2}$ для каждого $y \in \mathbb{F}_2^n$ [1]. Для множества бент-функций от n переменных используется обозначение $\mathcal{B}_n$. Для каждой $f \in \mathcal{B}_n$ однозначным образом из соотношения $W_f(y) = (-1)^{\widetilde{f}(y)} 2^{n/2}$ определяется *дуальная* к ней бент-функция $\widetilde{f} \in \mathcal{B}_n$. Бент-функция f называется *самодуальной* (*анти-самодуальной*), если $f = \widetilde{f}$ (соответственно $f = \widetilde{f} \oplus 1$). Множества самодуальных и анти-самодуальных бент-функций от n переменных обозначаются через $\mathrm{SB}^+(n)$ и $\mathrm{SB}^-(n)$ соответственно [2].

Открытой проблемой является полная характеризация и описание класса самодуальных бент-функций. Этому и другим вопросам, связанным с самодуальными бент-функциями, посвящён ряд работ (C. Carlet, L. E. Danielson, M. G. Parker, P. Solé, X. Hou, T. Feulner, L. Sok, A. Wassermann и др.). В частности, в работе [3] приведена аффинная классификация самодуальных бент-функций от $2, 4, 6$ переменных и всех квадратичных самодуальных бент-функций от 8 переменных относительно преобразования, сохраняющего самодуальность. В [2] дана классификация всех квадратичных самодуальных бент-функций. Аффинную классификацию квадратичных и кубических самодуальных бент-функций от 8 переменных относительно преобразования, сохраняющего самодуальность, можно найти в [4]. Верхняя оценка количества самодуальных бент-функций приведена в [5]. В работах [6 – 8] представлены конструкции самодуальных бент-функций. Связь самодуальных кватернарных бент-функций и самодуальных булевых бент-функций отмечена в [9].

Согласно [10], назовём ортогональной группой порядка n над полем $\mathbb{F}_2$ группу

$$\mathcal{O}_n = \left\{ L \in \mathrm{GL}\,(n, \mathbb{F}_2) : LL^{\mathrm{T}} = I_n \right\},$$

где L^{T} — транспонирование L; I_n — единичная матрица порядка n над полем $\mathbb{F}_2$.

Далее представлены известные результаты, касающиеся метрических свойств самодуальных бент-функций, опубликованные в работах [11 – 16].

1. Самодуальные бент-функции Мэйорана — МакФарланда

Бент-функции от $2k$ переменных, представимые в виде

$$f(x, y) = \langle x, \pi(y) \rangle \oplus g(y), \quad x, y \in \mathbb{F}_2^k,$$

где π — перестановка на множестве $\mathbb{F}_2^k$ и g — булева функция от k переменных, образуют известный класс *Мэйорана — МакФарланда* [17]. Данный класс имеет мощность $2^k! \cdot 2^{2^k}$.

Через $\mathrm{SB}^+_{\mathcal{M}}(n)$ ($\mathrm{SB}^-_{\mathcal{M}}(n)$) обозначим множество самодуальных (анти-самодуальных) бент-функций от n переменных из класса Мэйорана — МакФарланда. В работе [3] найдены необходимые и достаточные условия самодуальности бент-функций из класса Мэйорана — МакФарланда, а именно: бент-функция $f(x, y)$ Мэйорана — МакФарланда принадлежит множеству $\mathrm{SB}^+_{\mathcal{M}}(2k)$ тогда и только тогда, когда

$$\pi(y) = L(y \oplus c), \quad g(y) = \langle c, y \rangle \oplus d, \quad y \in \mathbb{F}_2^k,$$

где $L \in \mathcal{O}_k$; $c \in \mathbb{F}_2^k$; $\mathrm{wt}(c)$ — чётное число; $d \in \mathbb{F}_2^n$. Заметим, что $\left|\mathrm{SB}^+_{\mathcal{M}}(2k)\right| = 2^k \cdot |\mathcal{O}_k|$.

Всюду далее предполагаем, что n — чётное натуральное число. В [11] исследованы возможные расстояния Хэмминга между самодуальными бент-функциями из класса Мэйорана — МакФарланда.

Теорема 1 [11]. Пусть $n \geqslant 4$ и $f, g \in \mathrm{SB}^+_{\mathcal{M}}(n) \cup \mathrm{SB}^-_{\mathcal{M}}(n)$, тогда

$$\mathrm{dist}(f,g) \in \left\{ 2^{n-1}, 2^{n-1}\left(1 \pm \frac{1}{2^r}\right), r = 0, 1, \ldots, n/2-1 \right\}.$$

Более того, если $f, g \in \mathrm{SB}^+_{\mathcal{M}}(n)$ или $f, g \in \mathrm{SB}^-_{\mathcal{M}}(n)$, то все приведённые расстояния, кроме 2^{n-1}, являются достижимыми. Для произвольной пары функций $f \in \mathrm{SB}^+_{\mathcal{M}}(n)$ и $g \in \mathrm{SB}^-_{\mathcal{M}}(n)$ справедливо $\mathrm{dist}(f,g) = 2^{n-1}$.

Анализ приведённых расстояний позволяет вычислить минимальное расстояние Хэмминга между рассматриваемыми функциями.

Следствие 1. Пусть $n \geqslant 4$, тогда минимальное расстояние Хэмминга между самодуальными бент-функциями от n переменных из класса Мэйорана — МакФарланда равно 2^{n-2}.

В силу того, что минимальное расстояние Хэмминга между квадратичными булевыми функциями от n переменных (кодовыми словами кода Рида — Маллера $\mathrm{RM}(2,n)$) не меньше чем 2^{n-2} [18], получаем следующее

Следствие 2. Пусть $n \geqslant 4$, тогда минимальное расстояние Хэмминга между квадратичными булевыми функциями достижимо на самодуальных бент-функциях от n переменных из класса Мэйорана — МакФарланда.

2. Метрическая регулярность

Известно [19], что минимальное расстояние Хэмминга между бент-функциями от n переменных равно $2^{n/2}$. В работе [12] доказано, что при $n \geqslant 4$ данное расстояние достижимо на множестве (анти-)самодуальных бент-функций.

Утверждение 1 [12]. Пусть $n \geqslant 4$, тогда минимальное расстояние Хэмминга между (анти-)самодуальными бент-функциями от n переменных равно $2^{n/2}$.

Пусть $A \subseteq \mathbb{F}_2^n$ — произвольное множество и $y \in \mathbb{F}_2^n$ — произвольный двоичный вектор. Расстояние от вектора y до множества A определяется как $\mathrm{dist}(y,A) = \min\limits_{x \in A} \mathrm{dist}(y,x)$. *Радиусом покрытия* множества A называется число $\mathrm{d}(A) = \max\limits_{y \in \mathbb{F}_2^n} \mathrm{dist}(y,A)$. Множество двоичных векторов, находящихся на расстоянии $\mathrm{d}(A)$ от множества $A \subseteq \mathbb{F}_2^n$, называется *метрическим дополнением* множества A и обозначается $\widehat{A}$ [20]. Если $\widehat{\widehat{A}} = A$, то множество A называется *метрически регулярным*.

Рассматривая данные определения применительно к векторам значений булевых функций, можно определить *радиус покрытия*, *метрическое дополнение* и *метрическую регулярность* произвольного подмножества $M \subseteq \mathcal{F}_n$ [21].

В [3] доказано, что радиус покрытия множества $\mathrm{SB}^+(n)$ равен 2^{n-1}. Следующее утверждение описывает метрическое дополнение множества самодуальных бент-функций.

Теорема 2 [12]. Пусть $n \geqslant 4$, тогда булева функция от n переменных:

— является самодуальной бент-функцией в том и только в том случае, когда она находится на расстоянии 2^{n-1} от множества всех анти-самодуальных бент-функций от n переменных, то есть является элементом множества $\widehat{\mathrm{SB}^-(n)}$;

— является анти-самодуальной бент-функцией в том и только в том случае, когда она находится на расстоянии 2^{n-1} от множества всех самодуальных бент-функций от n переменных, то есть является элементом множества $\widehat{\mathrm{SB}^+(n)}$.

В [22] доказано, что аффинными являются булевы функции, которые находятся на максимально возможном удалении от множества бент-функций, что влечёт *дуальность* в определении аффинных функций и бент-функций. Таким образом, на основании теоремы 2 можно говорить о том, что между множествами самодуальных и анти-самодуальных бент-функций от $n \geqslant 4$ переменных существует метрическая *дуальность*.

На основании теоремы 2 (случай $n = 2$ рассмотрен отдельно) показано, что

Следствие 3 [12].

1) Множество $\mathrm{SB}^+(n)$ всех самодуальных бент-функций от n переменных является метрически регулярным.
2) Множество $\mathrm{SB}^-(n)$ всех анти-самодуальных бент-функций от n переменных является метрически регулярным.

3. Группа автоморфизмов

Отображение всех булевых функций от n переменных в себя называется *изометричным*, если оно сохраняет расстояние Хэмминга между каждой парой булевых функций от n переменных. Множество изометричных отображений множества всех булевых функций от n переменных в себя будем обозначать через $\mathcal{I}_n$. Известно, что каждое такое отображение однозначно представляется в виде

$$f(x) \longrightarrow f(\pi(x)) \oplus g(x),$$

где π — перестановка на множестве $\mathbb{F}_2^n$; g — булева функция от n переменных [23]. Отображение такого вида обозначим через $\varphi_{\pi,g} \in \mathcal{I}_n$. Известно, что каждое изометричное отображение множества всех булевых функций от чётного числа переменных n в себя, оставляющее множество $\mathcal{B}_n$ на месте, представимо в виде композиции аффинного преобразования координат и прибавления аффинной функции от n переменных [24].

Группой автоморфизмов фиксированного подмножества $M \subseteq \mathcal{F}_n$ называется группа элементов множества $\mathcal{I}_n$, оставляющая множество M на месте; она обозначается $\mathrm{Aut}\,(M)$.

В [4] (см. также [3]) доказано, что отображение всех булевых функций от n переменных в себя, имеющее вид

$$f(x) \longrightarrow f\left(L\left(x \oplus c\right)\right) \oplus \langle c, x\rangle \oplus d,$$

где $L \in \mathcal{O}_n$; $c \in \mathbb{F}_2^n$; $\mathrm{wt}(c)$ — чётное число; $d \in \mathbb{F}_2$, сохраняет самодуальность бент-функции. Нетрудно видеть, что все отображения данного вида являются элементами множества $\mathcal{I}_n$. Группа таких преобразований называется *расширенной ортогональной группой* и обозначается $\overline{\mathcal{O}}_n$ [4, 25]. Известно, что $\overline{\mathcal{O}}_n$ является подгруппой группы $\mathrm{GL}\,(n+2, \mathbb{F}_2)$ [4].

В [2] отмечено, что отображение всех булевых функций от n переменных в себя, имеющее вид

$$f(x) \longrightarrow f\left(x \oplus c\right) \oplus \langle c, x\rangle\,,$$

где $c \in \mathbb{F}_2^n$; $\mathrm{wt}(c)$ — нечётное число, определяет биекцию между множествами $\mathrm{SB}^+(n)$ и $\mathrm{SB}^-(n)$. Очевидно, что такое отображение сохраняет расстояние Хэмминга. Частный случай отображения данного вида — при $c = (1, 0, 0, \ldots, 0) \in \mathbb{F}_2^n$ — ранее был расмотрен в [3], на основании чего был сделан вывод о том, что между множествами $\mathrm{SB}^+(n)$ и $\mathrm{SB}^-(n)$ существует взаимно однозначное соответствие.

В [13] получено обобщение данных результатов в рамках класса изометричных отображений: доказано, что группы автоморфизмов множеств $\mathrm{SB}^+(n)$ и $\mathrm{SB}^-(n)$ совпадают.

Теорема 3 [13]. При $n \geqslant 4$ справедливо $\mathrm{Aut}\left(\mathrm{SB}^+(n)\right) = \mathrm{Aut}\left(\mathrm{SB}^-(n)\right)$.

Получен следующий критерий сохранения самодуальности.

Теорема 4 [13]. Пусть $n \geqslant 4$, тогда изометричное отображение $\varphi_{\pi,g}$ является элементом грппы $\mathrm{Aut}\left(\mathrm{SB}^+(n)\right)$ в том и только в том случае, когда для любых $x, y \in \mathbb{F}_2^n$ справедливо

$$\langle \pi(x), y\rangle \oplus g(x) = \left\langle x, \pi^{-1}(y)\right\rangle \oplus g\left(\pi^{-1}(y)\right).$$

С использованием этого критерия и теоремы 3 получено описание группы автоморфизмов множества (анти-)самодуальных бент-функций от n переменных.

Теорема 5 [13]. При $n \geqslant 4$ справедливо

$$\mathrm{Aut}\left(\mathrm{SB}^+(n)\right) = \mathrm{Aut}\left(\mathrm{SB}^-(n)\right) = \overline{\mathcal{O}}_n.$$

Из этих результатов следует, что более общего подхода к классификации самодуальных бент-функций на основе изометричных отображений, чем предложенный в [3, 4], не существует.

Применительно к биекциям между множествами $\mathrm{SB}^+(n)$ и $\mathrm{SB}^-(n)$ получен следующий критерий.

Теорема 6 [13]. Пусть $n \geqslant 4$, тогда изометричное отображение $\varphi_{\pi,g}$ определяет биекцию между множествами $\mathrm{SB}^+(n)$ и $\mathrm{SB}^-(n)$ в том и только в том случае, когда для любых $x, y \in \mathbb{F}_2^n$ справедливо

$$\langle \pi(x), y\rangle \oplus g(x) = \left\langle x, \pi^{-1}(y)\right\rangle \oplus g\left(\pi^{-1}(y)\right) \oplus 1.$$

С использованием данного критерия получена общая форма изметричных отображений, определяющих биекцию между множествами $\mathrm{SB}^+(n)$ и $\mathrm{SB}^-(n)$.

Теорема 7 [13]. При $n \geqslant 4$ изометричное отображение $\varphi_{\pi,g} \in \mathcal{I}_n$ определяет биекцию между множествами $\mathrm{SB}^+(n)$ и $\mathrm{SB}^-(n)$, если и только если

$$\pi(x) = L\left(x \oplus c\right),\ g(x) = \langle c, x\rangle \oplus d,\ x \in \mathbb{F}_2^n,$$

где $L \in \mathcal{O}_n$; $c \in \mathbb{F}_2^n$; $\mathrm{wt}(c)$ — чётное число; $d \in \mathbb{F}_2$.

Из теорем 5 и 7 следует, что чётность веса Хэмминга вектора $c \in \mathbb{F}_2^n$, фигурирующего в описании расширенной ортогональной группы, является «переключателем» между изометричным отображением, сохраняющим (анти-)самодуальность, и изометричным отображением, меняющим местами самодуальные и анти-самодуальные бент-функции.

4. Расстояние Хэмминга между бент-функций и дуальной к ней

Согласно [3, 25], *отношением Рэлея* (the Rayleigh quotient) S_f булевой функции f от n переменных называется число

$$S_f = \sum_{x,y\in\mathbb{F}_2^n} (-1)^{f(x)\oplus f(y)\oplus\langle x,y\rangle} = \sum_{y\in\mathbb{F}_2^n} (-1)^{f(y)} W_f(y).$$

Известно [3], что абсолютное значение S_f не превосходит $2^{3n/2}$, при этом в случае, когда n — чётное число, данная оценка достигается только на самодуальных бент-функциях $\left(+2^{3n/2}\right)$ и анти-самодуальных бент-функциях $\left(-2^{3n/2}\right)$.

В [13] исследованы вопросы сохранения и смены знака отношения Рэлея каждой булевой функции от n переменых при изометричных преобразованиях.

Теорема 8 [13]. Пусть $n \geqslant 4$, тогда изометричное отображение $\varphi_{\pi,g} \in \mathcal{I}_n$ сохраняет отношение Рэлея каждой булевой функции от n переменных в том и только в том случае, когда $\varphi_{\pi,g} \in \mathrm{Aut}\left(\mathrm{SB}^+(n)\right)$.

Теорема 9 [13]. Пусть $n \geqslant 4$, тогда изометричное отображение $\varphi_{\pi,g} \in \mathcal{I}_n$ меняет знак отношения Рэлея каждой булевой функции от n переменных в том и только в том случае, когда оно определяет биекцию между множествами $\mathrm{SB}^+(n)$ и $\mathrm{SB}^-(n)$.

Пусть $f \in \mathcal{B}_n$. Из соотношения

$$\mathrm{dist}(f, \widetilde{f}) = 2^{n-1} - \frac{S_f}{2^{n/2+1}}$$

следует, что отношение Рэлея полностью характеризует расстояние Хэмминга между бент-функцией $f \in \mathcal{B}_n$ и дуальной к ней функцией $\widetilde{f} \in \mathcal{B}_n$. Таким образом, на основе теорем 5 и 8 можно получить следующий результат.

Теорема 10 [13]. При $n \geqslant 4$ изометричное отображение $\varphi_{\pi,g} \in \mathcal{I}_n$ оставляет множество бент-функций от n переменных на месте и сохраняет расстояние Хэмминга между бент-функцией и дуальной к ней тогда и только тогда, когда

$$f(x) \longrightarrow f\left(L\left(x \oplus c\right)\right) \oplus \langle c, x\rangle \oplus d, \quad x \in \mathbb{F}_2^n,$$

где $L \in \mathcal{O}_n$; $c \in \mathbb{F}_2^n$; $\mathrm{wt}(c)$ — чётное число; $d \in \mathbb{F}_2$.

ЛИТЕРАТУРА

1. *Rothaus O.* On bent functions // J. Combin. Theory. Ser. A. 1976. V. 20. No. 3. P. 300–305.
2. *Hou X.-D.* Classification of self dual quadratic bent functions // Des. Codes Cryptogr. 2012. V. 63. No. 2. P. 183–198.
3. *Carlet C., Danielson L. E., Parker M. G., and Solé P.* Self-dual bent functions // Int. J. Inform. Coding Theory. 2010. V. 1. P. 384–399.
4. *Feulner T., Sok L., Solé P., and Wassermann A.* Towards the classification of self-dual bent functions in eight variables // Des. Codes Cryptogr. 2013. V. 68. No. 1. P. 395–406.
5. *Hyun J. Y., Lee H., and Lee Y.* MacWilliams duality and Gleason-type theorem on self-dual bent functions // Des. Codes Cryptogr. 2012. V. 63. No. 3. P. 295–304.
6. *Luo G., Cao X., and Mesnager S.* Several new classes of self-dual bent functions derived from involutions // Cryptogr. Commun. 2019. V. 11. No. 6. P. 1261–1273.
7. *Mesnager S.* Several new infinite families of bent functions and their duals // IEEE Trans. Inf. Theory. 2014. V. 60. No. 7. P. 4397–4407.
8. *Rifà J. and Zinoviev V. A.* On binary quadratic symmetric bent and almost bent functions. arXiv:1211.5257v3, 2019.
9. *Sok L., Shi M., and Solé P.* Classification and Construction of quaternary self-dual bent functions // Cryptogr. Commun. 2018. V. 10. No. 2. P. 277–289.
10. *Janusz G. J.* Parametrization of self-dual codes by orthogonal matrices // Finite Fields Appl. 2007. V. 13. No. 3. P. 450–491.
11. *Kutsenko A. V.* The Hamming distance spectrum between self-dual Maiorana — McFarland bent functions // J. Appl. Industr. Math. 2018. V. 12. No. 1. P. 112–125.

12. *Kutsenko A.* Metrical properties of self-dual bent functions // Des. Codes Cryptogr. 2020. V. 88. No. 1. P. 201–222.
13. *Kutsenko A.* The group of automorphisms of the set of self-dual bent functions // Cryptogr. Commun. 2020.
14. *Куценко А. В.* О множестве расстояний Хэмминга между самодуальными бент-функциями // Прикладная дискретная математика. Приложение. 2016. № 9. С. 29–30.
15. *Куценко А. В.* О некоторых свойствах самодуальных бент-функций // Прикладная дискретная математика. Приложение. 2018. № 11. С. 44–46.
16. *Куценко А. В.* Изометричные отображения множества всех булевых функций в себя, сохраняющие самодуальность и отношение Рэлея // Прикладная дискретная математика. Приложение. 2019. № 12. С. 55–58.
17. *McFarland R. L.* A family of difference sets in non-cyclic groups // J. Combin. Theory. Ser. A. 1973. V. 15. No. 1. P. 1–10.
18. *MacWilliams F. J. and Sloane N. J. A.* The Theory of Error-Correcting Codes. Amsterdam, New York, Oxford, North-Holland, 1983. 782 p.
19. *Коломеец Н. А., Павлов А. В.* Свойства бент-функций, находящихся на минимальном расстоянии друг от друга // Прикладная дискретная математика. 2009. № 4. С. 5–20.
20. *Облаухов А. К.* О метрическом дополнении подпространств булева куба // Дискретный анализ и исследование операций. 2016. Вып. 23. № 3. С. 93–106
21. *Tokareva N.* Bent Functions: Results and Applications to Cryptography. Acad. Press, Elsevier, 2015. 230 p.
22. *Tokareva N.* Duality between bent functions and affine functions // Discrete Math. 2012. V. 312. No. 3. P. 666–670.
23. *Марков А. А.* О преобразованиях, не распространяющих искажения // Избранные труды. Т. II. Теория алгорифмов и конструктивная математика, математическая логика, информатика и смежные вопросы. М.: МЦНМО, 2003. С. 70–93.
24. *Tokareva N. N.* The group of automorphisms of the set of bent functions // Discrete Math. Appl. 2010. V. 20. No. 5. P. 655–664.
25. *Danielsen L. E., Parker M. G., and Solé P.* The Rayleigh quotient of bent functions // LNCS. 2009. V. 5921. P. 418–432.

УДК 519.7 DOI 10.17223/2226308X/13/6

КРИПТОГРАФИЧЕСКИЕ СВОЙСТВА НЕКОТОРЫХ КОМПОЗИЦИЙ ВЕКТОРНЫХ БУЛЕВЫХ ФУНКЦИЙ

Е. С. Липатова

Рассматриваются три класса обратимых векторных булевых функций, таких, что каждая их координатная функция существенно зависит от заданного числа переменных. Приведены результаты экспериментального исследования криптографических свойств композиций функций из этих классов.

Ключевые слова: *векторная булева функция, нелинейность, алгебраическая иммунность, дифференциальная равномерность.*

Обозначим через $\mathcal{F}_n$ множество всех подстановок на $\mathbb{F}_2^n$ и будем рассматривать следующие подклассы функций из $\mathcal{F}_n$:

1) $\mathcal{K}_n$ — функции, полученные из тождественной подстановки с помощью n независимых транспозиций [1];

2) $\mathcal{S}_{n,k}$ — функции вида $F = (f_1, \ldots, f_n)$, где $(f_1, \ldots, f_k) \in \mathcal{K}_k$ и $f_i(x_1, \ldots, x_n) = x_i \oplus \oplus\, g_i(x_1, \ldots, x_{i-1})$, g_i — произвольные булевы функции, существенно зависящие от $k-1$ переменных, $i = k+1, \ldots, n$, $k \leqslant n$ [2, 3];
3) пусть $k|n$; $s = n/k$; $\mathcal{P}_{n,k}$ — функции $F = (f_1, \ldots, f_n)$, где $f_{tk+i}(x_1, \ldots, x_n) = g_i^{(t+1)}(x_{tk+1}, \ldots, x_{(t+1)k})$, $t = 0, \ldots, s-1$ и $i = 1, \ldots, k$; $\left(g_1^{(j)}, \ldots, g_k^{(j)}\right) \in \mathcal{K}_k$, $j = 1, \ldots, s$.

Приведём определения некоторых криптографических характеристик функций $F = (f_1 \ldots f_n) : \mathbb{F}_2^n \to \mathbb{F}_2^n$ [4–6].

Компонентой функции F называется булева функция $vF = v_1 f_1 \oplus \ldots \oplus v_n f_n$, где $v = v_1 \ldots v_n \in \mathbb{F}_2^n \setminus \{0^n\}$; 0^n — нулевой вектор длины n.

Нелинейностью $N(F)$ и *компонентной алгебраической иммунностью* $\mathrm{AI}_{\mathrm{comp}}(F)$ функции F называются минимальные нелинейность и алгебраическая иммунность её компонент соответственно:

$$N(F) = \min_{v \in \mathbb{F}_2^m \setminus \{0^m\}} N(vF), \quad \mathrm{AI}_{\mathrm{comp}}(F) = \min_{v \in \mathbb{F}_2^m \setminus \{0^m\}} \mathrm{AI}(vF).$$

Для векторов $a, b \in \mathbb{F}_2^n$ обозначим $\delta_F(a,b) = |\{x \in \mathbb{F}_2^n : F(x) \oplus F(x \oplus a) = b\}|$. *Показателем дифференциальной равномерности* функции F называется

$$\delta_F = \max_{a \neq 0^n, b} \delta_F(a,b).$$

Проведено экспериментальное исследование этих характеристик, а также алгебраической степени для некоторых композиций функций от 3–10 переменных. Нелинейность, алгебраическая иммунность, дифференциальная равномерность вычислялись по алгоритмам, описанным в [7]; алгебраическая степень — с помощью преобразования Мёбиуса.

В ходе экспериментов получены следующие результаты:

1) Если в композиции участвует случайная подстановка $F \in \mathcal{F}_n$, то характеристики композиции в среднем совпадают с характеристиками функции F. Это можно объяснить тем, что при композиции любой подстановки со случайной равновероятно выбранной равномерное распределение сохраняется [8].
2) Композиция H функции $F \in \mathcal{K}_n$ с функцией $G \in \mathcal{S}_{n,k} \cup \mathcal{P}_{n,k}$ (в любом порядке), можно сказать, берёт наилучшие свойства обоих классов:
   а) $\deg H$ и $\mathrm{AI}_{\mathrm{comp}}(H)$ сохраняются, как у функции F;
   б) δ_H принимает значения, приблизительно равные δ_F, если $G \in \mathcal{S}_{n,k}$, и δ_G, если $G \in \mathcal{P}_{n,k}$;
   в) $N(H) \approx N(G)$.
3) При композиции функций F и G из одного класса, $\mathcal{K}_n$ или $\mathcal{P}_{n,k}$, все рассмотренные свойства в общем ухудшаются:
   а) δ_H принимает худшее значение 2^n, а нелинейность — значение 0 (хотя очень редко получаются значения лучше, чем у исходных функций);
   б) $\mathrm{AI}_{\mathrm{comp}}(H)$ принимает значения 1 или 2 (у исходных функций всегда 2);
   в) $\deg H$ часто равна 1 (редко — как у исходных функций).
4) Про композиции функций из разных классов $\mathcal{S}_{n,k}$ и $\mathcal{P}_{n,k}$ и композиции функций одного класса $\mathcal{S}_{n,k}$ нельзя однозначно сказать о поведении свойств функций:
   а) $\deg H$ часто улучшается;
   б) $\mathrm{AI}_{\mathrm{comp}}(H)$ принимает значения 1 или 2 (те же значения, что и у функций класса $\mathcal{S}_{n,k}$);

в) δ_H в первом случае принимает значения, приблизительно равные δ функций классов $\mathcal{P}_{n,k}$ (очень редко получаются значения лучше или хуже), а во втором случае сохраняется значение 2^n (как у функций класса $\mathcal{S}_{n,k}$);

г) $N(H)$ в обоих случаях может принимать разные значения, в основном они лучше, чем наихудшие значения у функций, используемых в композициях.

ЛИТЕРАТУРА

1. *Pankratova I. A.* Construction of invertible vectorial Boolean functions with coordinates depending on given number of variables // Материалы Междунар. науч. конгресса по информатике: Информационные системы и технологии. Республика Беларусь, Минск, 24–27 окт. 2016. Минск: БГУ, 2016. С. 519–521.
2. *Agibalov G. P.* Substitution block ciphers with functional keys // Прикладная дискретная математика. 2017. № 38. С. 57–65.
3. *Панкратова И. А.* Об обратимости векторных булевых функций // Прикладная дискретная математика. Приложение. 2015. № 8. С. 35–37.
4. *Carlet C.* Vectorial Boolean Functions for Cryptography. Cambridge: Cambridge University Press, 2010. 93 p.
5. *Canteaut A.* Lecture Notes on Cryptographic Boolean Functions. Paris: Inria, 2016. 48 p.
6. *Nyberg K.* Differentially uniform mappings for cryptography // LNCS. 1994. V. 765. P. 55–64.
7. *Киселева Н. М., Липатова Е. С., Панкратова И. А., Трифонова Е. Е.* Алгоритмы вычисления криптографических характеристик векторных булевых функций // Прикладная дискретная математика. 2019. № 46. С. 78–87.
8. *Кнут Д.* Искусство программирования. Т. 2. Получисленные алгоритмы. М.: Вильямс, 2007. 832 с.

УДК 519.7 DOI 10.17223/2226308X/13/7

КРИПТОГРАФИЧЕСКИЕ СВОЙСТВА ОРТОМОРФИЗМОВ[1]

Ю. П. Максимлюк

Рассмотрены взаимно однозначные отображения $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, называемые ортоморфизмами, такие, что отображения $G(x) = F(x) \oplus x$ также являются взаимно однозначными. Они используются в схеме Лая — Месси в качестве перемешивающего элемента между раундами, а также для построения криптографически стойких S-блоков. Исследованы основные криптографические свойства: нелинейные характеристики и дифференциальная равномерность. Выявлено, что ортоморфизмы от малого числа переменных не устойчивы к линейному и дифференциальному криптоанализам.

Ключевые слова: *ортоморфизм, таблица линейного преобладания, таблица дифференциалов.*

В симметричной криптографии часто используются отображения множества $\mathbb{Z}_2^n$, состоящего из двоичных наборов длины n, на себя. В частности, в [1] в шифрах FOX (IDEA NXT), использующих схему Лая — Месси, предлагается использовать отображение, называемое ортоморфизмом.

[1]Работа выполнена при поддержке Математического центра в Академгородке (г. Новосибирск), соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2019-1613, и Лаборатории криптографии JetBrains Research.

Ортоморфизм $\mathbb{Z}_2^n$ — это взаимно однозначное отображение $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$, такое, что отображение $G(x) = F(x) \oplus x$ также является взаимно однозначным.

В литературе в основном освещаются перемешивающие свойства ортоморфизмов. Например, в [2] ортоморфизмы характеризуются свойством отображать каждую максимальную подгруппу группы двоичных наборов длины n наполовину в себя и наполовину в своё дополнение.

В рамках данной работы разработан и программно реализован рекурсивный алгоритм построения всех ортоморфизмов для заданного n. Алгоритм перебирает все значения для k-го элемента и проверяет выполнение определения ортоморфизма. Если проверка успешна, то переходим к $(k+1)$-му элементу, иначе проверяем следующее значение k-го. Когда проверены все возможные значения для k-й позиции, происходит возврат к дальнейшей проверке значений для позиции $k-1$.

С помощью этой программы получены все ортоморфизмы для малых значений n и один ортоморфизм для $n = 16$ для исследования модификации шифра Simon 32/64 [3], где вместо сети Фейстеля использована схема Лая — Месси.

Получено, что:

— при $n = 2$ существует 8 ортоморфизмов;
— при $n = 3$ существует 384 ортоморфизма;
— при $n = 4$ существует 244744192 ортоморфизма.

Для всех полученных ортоморфизмов экспериментально исследованы основные криптографические свойства: нелинейные характеристики и дифференциальная равномерность.

Обозначим вход и выход функции $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ через $x = (x_1, \ldots, x_n)$ и $y = (y_1, \ldots, y_n)$ соответственно. Для линейного криптоанализа строится таблица линейного преобладания, где на пересечении строки $u \in \mathbb{Z}_2^n$ и столбца $v \in \mathbb{Z}_2^n$ находится число λ, такое, что соотношение $\langle u, x\rangle = \langle v, y\rangle$ выполняется с вероятностью $(2^{n-1} + \lambda)/2^n$, где $\langle u, x\rangle = u_1x_1 \oplus \ldots \oplus u_nx_n$.

Утверждение 1. При $n = 2, 3$ и 4 таблицы линейного преобладания ортоморфизмов состоят из значений 0 и $\pm 2^{n-1}$.

Для дифференциального криптоанализа в таблице дифференциалов на пересечении строки $u \in \mathbb{Z}_2^n$ и столбца $v \in \mathbb{Z}_2^n$ находится число λ, такое, что равенство $F(x \oplus u) \oplus F(x) = v$ выполняется в точности для λ различных x.

Утверждение 2. При $n = 2, 3$ и 4 таблицы дифференциалов ортоморфизмов состоят из значений 0 и 2^n.

Для полученного ортоморфизма при $n = 16$ также исследовались таблицы линейного преобладания и дифференциалов. Таблица линейного преобладания состоит из значений 0 и $\pm 2^{n-1}$, а таблица дифференциалов — из 0 и 2^n.

Утверждения 1, 2 и точечное исследование ортоморфизма для $n = 16$ позволяют предположить, что для любого значения n таблицы дифференциалов и линейного преобладания ортоморфизмов имеют вид, описанный выше. Из этого следует, что ортоморфизмы не устойчивы к линейному и дифференциальному криптоанализам и должны использоваться в шифрах в качестве вспомогательных элементов для построения более устойчивых к криптоанализу перемешивающих отображений.

ЛИТЕРАТУРА

1. *Nakahara J. Jr.* Lai-Massey Cipher Designs. History, Design Criteria and Cryptanalysis. Springer, 2018. 726 p.

2. *Mittenthal L.* Block substitutions using orthomorphic mappings // Adv. Appl. Math. 1995. V. 16. Iss. 1. P. 59–71.
3. *Beaulieu R., Shors D., Smith J., et al.* The Simon and Speck Families Of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, 2013.

УДК 519.7 DOI 10.17223/2226308X/13/8

О РАЗЛОЖЕНИИ ВЕКТОРНОЙ БУЛЕВОЙ ФУНКЦИИ В КОМПОЗИЦИЮ ДВУХ ВЕКТОРНЫХ ФУНКЦИЙ[1]

Г. М. Пинтус

Исследуется возможность представления векторной булевой функции в виде композиции двух векторных булевых функций меньшей алгебраической степени. Вводится понятие разложимости векторной булевой функции. Изучен вопрос сохранения разложимости при расширенном аффинном преобразовании. Представлена конструкция векторной булевой функции третьей степени от произвольного числа переменных, являющейся разложимой. Проведён вычислительный эксперимент, в результате которого показано, что все кубические векторные булевы функции от трёх переменных являются разложимыми.

Ключевые слова: *векторная булева функция, декомпозиция, пороговая реализация.*

Атаки по сторонним каналам [1] — вид атак, целью которых является нахождение уязвимостей в реализации криптографической системы. На данный момент эти атаки являются одними из наиболее эффективных среди всех видов криптоанализа. В атаках по сторонним каналам используется информация, полученная при отслеживании перепадов напряжений, времени выполнения процессов, электромагнитного излучения или звуков при проводимых алгоритмом вычислениях.

Пороговая реализация [2] является контрмерой по отношению к атакам по сторонним каналам, она разделяет наборы входных данных и используемые векторные булевы функции на части, позволяя скрыть различия между операциями. Таким образом, если разбиение удовлетворяет ряду условий, при работе алгоритма не происходит утечки информации, которая может быть использована в атаке по сторонним каналам.

В данном методе необходимо построить разбиение для векторной булевой функции определённым образом, что не всегда удаётся сделать. Однако придуман способ решения проблемы, использующий построение разбиения для более простых функций, композицией которых является исходная векторная булева функция.

В настоящей работе анализируется возможность представления векторных булевых функций в виде композиции векторных булевых функций меньших степеней. Рассмотрены векторные булевы функции от трёх переменных с алгебраической степенью равной трём и возможность их представления в виде композии двух векторных булевых функций алгебраической степени два.

Так как важно сохранение свойств при преобразованиях, а одним из наиболее распространённых является расширенное аффинное преобразование, мы исследуем вопрос сохранения разложимости векторной булевой функции при расширенной аффинной эквивалентности.

Векторной булевой функцией ((n, m)-функцией) F называется произвольное отображение $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$. В случае $m = 1$ говорят, что F — *булева функция от* n *пере-*

[1]Работа выполнена при поддержке Лаборатории криптографии JetBrains Research.

менных; (n,m)-функция F может быть задана набором из m координатных булевых функций от n переменных: $F(x) = (f_1(x), f_2(x), \ldots, f_m(x))$, $x \in \mathbb{F}_2^n$. Любую (n,m)-функцию можно единственным образом записать в виде *полинома Жегалкина*, или *алгебраической нормальной формы* (АНФ):

$$F(x_1, \ldots, x_n) = \Big(\bigoplus_{k=1}^{n} \bigoplus_{i_1,\ldots,i_k} a_{i_1,\ldots,i_k} x_{i_1} \cdot \ldots \cdot x_{i_k} \Big) \oplus a_0,$$

где для каждого k индексы $i_1, \ldots, i_k$ попарно различны и множества $\{i_1, \ldots, i_k\}$ являются всеми различными непустыми подмножествами множества $\{1, \ldots, n\}$; коэффициенты $a_{i_1,\ldots,i_k}, a_0$ принимают значения из $\mathbb{F}_2^m$. *Алгебраической степенью* $\deg(F)$ функции F называется количество переменных в самом длинном слагаемом АНФ, коэффициент при котором не равен нулевому вектору. Функция степени не выше 1 называется *аффинной*, при этом в случае $a_0 = 0$ функция *линейна*.

Две (n,n)-функции F и G называются *расширенно аффинно эквивалентными* (EA-эквивалентными), если существуют две аффинные (n,n)-подстановки A, B на множестве $\mathbb{F}_2^n$ и аффинная (n,n)-функция C, такие, что $G(x) = (B \circ F \circ A)(x) + C(x)$, $x \in \mathbb{F}_2^n$.

Пусть F — такая (n,n)-функция, что существуют (n,n)-функции G, H, для которых $\max\{\deg(G), \deg(H)\} < \deg(F)$ и $F(x) = G(H(x))$ для всех $x \in \mathbb{F}_2^n$. Функцию F степени $d > 2$, допускающую такую декомпозицию, будем называть *разложимой*.

Теорема 1. Пусть (n,n)-функция F степени $d > 2$ разложима. Тогда (n,n)-функция $F' = A_2 \circ F \circ A_1$, где A_1, A_2 — произвольные аффинные (n,n)-подстановки, также разложима. Если F представима в виде композиции двух (n,n)-функций G, H степени меньше d, таких, что функция H обратима и $\deg(H^{-1}) \leqslant \max\{\deg(G), \deg(H)\}$, то (n,n)-функция $F'' = F + A_0$ разложима для любой аффинной (n,n)-функции A_0.

Получена конструкция, которая позволяет для любого n построить класс разложимых векторных булевых функций третьей степени.

Теорема 2. Пусть $i, j, p, q \in \{1, \ldots, n\}$, $i \neq j$, $p \neq q$; $\{l_k : k = 1, \ldots, n\}$ и $\{l'_r : r = 1, \ldots, n\}$ — наборы произвольных линейных булевых функций от n переменных, такие, что $\deg(x_p x_q (l_i(x) + l_j(x))) = 3$; $Y(x) = (y_1(x), \ldots, y_n(x))$, где $y_k(x) = x_p x_q + l_k(x)$ при $k = 1, \ldots, n$, $x \in \mathbb{F}_2^n$. Тогда разложимой является векторная булева функция $F(x)$, определённая следующим образом:

$$F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \ldots \\ f_n(x) \end{pmatrix} = \begin{pmatrix} x_p x_q (l_i(x) + l_j(x)) + x_p x_q + l_i(x) l_j(x) + l'_1(Y(x)) \\ x_p x_q (l_i(x) + l_j(x)) + x_p x_q + l_i(x) l_j(x) + l'_2(Y(x)) \\ \ldots \\ x_p x_q (l_i(x) + l_j(x)) + x_p x_q + l_i(x) l_j(x) + l'_n(Y(x)) \end{pmatrix}.$$

ЛИТЕРАТУРА

1. *Bhunia S. and Tehranipoor M.* Hardware Security. A Hands-On Learning Approach. Elsevier Inc., 2019. 526 p.
2. *Nikova S., Rechberger C., and Rijmen V.* Threshold implementations against side-channel attacks and glitches // Inform. Commun. Technol. 2006. No. 4307. P. 529—546.

УДК 519.7 DOI 10.17223/2226308X/13/9

ОЦЕНКА НЕЛИНЕЙНОСТИ СБАЛАНСИРОВАННЫХ БУЛЕВЫХ ФУНКЦИЙ, ПОРОЖДЁННЫХ ОБОБЩЁННОЙ КОНСТРУКЦИЕЙ ДОББЕРТИНА[1]

И. А. Сутормин

Предложено обобщение конструкции Доббертина для высоконелинейных сбалансированных булевых функций. Исследован спектр Уолша — Адамара и получены оценки спектрального радиуса предложенных функций. Доказана точная верхняя оценка на спектральный радиус (нижняя оценка нелинейности) и предложен способ построить сбалансированную функцию Θ от $2n$ переменных при помощи сбалансированной θ от $n-k$ переменных со спектральным радиусом $R_\Theta = 2^n + 2^k R_\theta$, где R_Θ и R_θ — спектральные радиусы Θ и θ соответственно.

Ключевые слова: *булевы функции, бент-функции, сбалансированность, нелинейность, спектральный радиус.*

В различных криптографических алгоритмах часто используются булевы функции. Нелинейность — одно из основных для них свойств, оно показывает, насколько хорошо функцию можно приблизить некоторой линейной функцией, работать с которой значительно проще. Шифр может стать уязвимым к линейному криптоанализу при низкой нелинейности даже одной его части. Примером криптографического алгоритма, скомпрометированного своими компонентами с низкой нелинейностью, может послужить старый стандарт шифрования США — DES.

Введём необходимые определения. *Преобразование Уолша — Адамара* булевой функции f определяется как $W_f(a) = \sum\limits_{x \in \mathbb{F}_2^n} (-1)^{f(x)+\langle x,a\rangle}$, $a \in \mathbb{F}_2^n$; *спектральный радиус* $R_f = \max\limits_{a \in \mathbb{F}_2^n} |W_f(a)|$ и *нелиненость* $N_f = 2^{n-1} - R_f/2$. *Бент-функциями* называются функции от чётного числа переменных с максимальной возможной нелинейностью. Они впервые описаны в [1]. Подробную информацию об этом классе функций можно найти в [2, 3]. Булевы функции f и g от n переменных *аффинно эквивалентны*, если для всех x выполнено $g(x) = f(Ax+b)$, где A — невырожденная матрица размера $n \times n$; b — вектор длины n.

В практических целях часто требуется чтобы функция была *сбалансированной* — принимала значения 0 и 1 на одном и том же числе аргументов. Но максимальное значение нелинейности сбалансированных функций неизвестно, начиная уже с восьми переменных. Лучшие оценки получаются как следствие конкретных конструкций сбалансированных функций.

Конструкция, описанная Доббертином в [4], основана на модификации нормальных бент функций — функций от $2n$ переменных, постоянных на некотором аффинном подпространстве L размерности n. Суть конструкции заключается в замене значений бент-функции на подпространстве L значениями сбалансированной функции θ от n переменных. При этом спектральный радиус получившейся сбалансированной функции Θ равен $R_\Theta = 2^n + R_\theta$, а её нелинейность — $N_\Theta = 2^{2n-1} - 2^{n-1} - R_\theta/2$. В [4] сформулирована не опровергнутая до сих пор гипотеза о несуществовании сбалансированных функций с нелинейностью выше, чем можно получить при помощи этой конструкции.

[1]Работа выполнена в рамках государственного задания Института математики им. С. Л. Соболева СО РАН (проект № 0314-2019-0017) при поддержке РФФИ (проект № 20-31-70043) и Лаборатории криптографии JetBrains Research.

Рассмотрим обобщение конструкции Доббертина, использующее бент-функции с близкими к нормальности свойствами, а именно бент-функции от $2n$ переменных, принимающие постоянное значение на нескольких сдвигах некоторого подпространства L размерности $n-k$, $0 \leqslant k \leqslant n-2$. Так как аффинная эквивалентность сохраняет нелинейность и сбалансированность, можно без ограничения общности рассматривать такие бент-функции в виде $f : \mathbb{F}_2^{n-k} \times \mathbb{F}_2^{n+k} \to \mathbb{F}_2$, для которой существуют подмножества $I_0, I_1 \subset \mathbb{F}_2^{n+k}$ мощностей $|I_0| = 2^{2k-1} + 2^{k-1}$, $|I_1| = 2^{2k-1} - 2^{k-1}$, для которых справедливо

$$f(x,y) \equiv 0 \ \text{ при } y \in I_0,$$
$$f(x,y) \equiv 1 \ \text{ при } y \in I_1.$$

Такое представление прямо связано с конструкцией вида $\widetilde{f} + \mathrm{Ind}_{L^\perp}$, подробную информацию о которой можно найти в [5–7]. Здесь $\widetilde{f}$ — дуальная к f функция [3].

При помощи бент-функции такого вида и набора θ_y, $y \in I_0 \cup I_1$, сбалансированных функций от $n-k$ переменных строится обобщающая конструкцию Доббертина функция Θ:

$$\Theta(x,y) = \begin{cases} \theta_y(x) & \text{при } y \in I_0 \cup I_1, \\ f(x,y) & \text{иначе.} \end{cases} \tag{1}$$

При $k = 0$ описанная конструкция полностью совпадает с конструкцией Доббертина, при $k = 1$ она также эквивалентна конструкции Доббертина.

Теорема 1. Функция Θ вида (1) является сбалансированной функцией и её коэффициенты Уолша — Адамара вычисляются по формуле

$$W_\Theta(a,b) = \begin{cases} W_f(a,b) + \sum\limits_{y \in I_0 \cup I_1} (-1)^{\langle b,y \rangle} W_{\theta_y}(a), & \text{если } a \neq 0, \\ 0 & \text{иначе.} \end{cases}$$

Следствие 1. Спектральный радиус Θ не превосходит $2^n + \sum\limits_{y \in I_0 \cup I_1} R_{\theta_y}$, причём всегда можно выбрать θ_y, при которых оценка достигается.

Теорема 2. Пусть θ — сбалансированная функция $n-k$ переменных, $\theta_y = \theta$ при $y \in I_0$ и $\theta_y = \theta \oplus 1$ при $y \in I_1$. Тогда

$$R_\Theta = 2^n + 2^k R_\theta.$$

Получившееся R_Θ зависит от R_θ, k и n. Несмотря на то, что θ является функцией от $n-k$ переменных, наилучший результат достигается при $k = 0$, то есть в случае, описанном Доббертином.

ЛИТЕРАТУРА

1. *Rothaus O.* On "bent" functions // J. Combin. Theory. Ser. A. 1976. V. 20. No. 3. P. 300–305.
2. *Логачев О. А., Сальников А. А., Смышляев С. В., Ященко В. В.* Булевы функции в теории кодирования и криптологии. 2-е изд. М.: МЦНМО, 2012. 584 с.
3. *Tokareva N. N.* Bent Functions. Results and Applications to Cryptography. Acad. Press. Elsevier, 2015.
4. *Dobbertin H.* Construction of bent functions and balanced Boolean functions with high nonlinearity // LNCS. 1994. V. 1008. P. 61–74.

5. *Kolomeec N.* On properties of a bent function secondary construction // Proc. BFA'2020. https://boolean.w.uib.no/bfa-2020.
6. *Коломеец Н. А.* О некоторых свойствах конструкции бент-функций с помощью подпространств произвольной размерности // Прикладная дискретная математика. Приложение. 2018. № 11. С. 41–43.
7. *Carlet C.* Two new classes of bent functions // LNCS. 1994. V. 765. P. 77–101.

УДК 519.7 DOI 10.17223/2226308X/13/10

СВЯЗЬ МЕЖДУ КВАТЕРНАРНЫМИ И КОМПОНЕНТНЫМИ БУЛЕВЫМИ БЕНТ-ФУНКЦИЯМИ[1]

А. С. Шапоренко

Исследуются кватернарные бент-функции. Функция $g : \mathbb{Z}_4^n \to \mathbb{Z}_4$ называется кватернарной функцией от n переменных. Доказано, что свойство кватернарной функции $g(x + 2y) = a(x, y) + 2b(x, y)$ быть бент напрямую не зависит от того, являются ли функции b и $a \oplus b$ булевыми бент-функциями. Получено количество кватернарных бент-функций от одной и двух переменных с описанием свойств булевых функций b и $a \oplus b$. Представлены простые конструкции кватернарных бент-функций от любого числа переменных.

Ключевые слова: *кватернарные функции, булевы функции, бент-функции.*

Пусть $\langle x, y\rangle$ обозначает скалярное произведение двоичных векторов x и y по модулю 2, а $x.y$ — их скалярное произведение по модулю 4.

Функция $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ называется *булевой функцией* от n переменных. *Преобразованием Уолша — Адамара булевой функции* f от n переменных называется целочисленная функция $W_f(x)$, заданная на множестве $\mathbb{Z}_2^n$ равенством

$$W_f(x) = \sum_{y \in \mathbb{Z}_2^n} (-1)^{\langle x,y\rangle \oplus f(y)}.$$

Булева функция f от чётного числа n переменных называется *бент-функцией*, если $|W_f(x)| = 2^{n/2}$ для любого $x \in \mathbb{Z}_2^n$.

Шифры, в которых используются бент-функции, более устойчивы к *линейному криптоанализу* [1], потому что бент-функции крайне плохо аппроксимируются аффинными функциями. Бент-функции используются в блочном шифре CAST как координатные функции S-блоков [2], а также для построения регистра сдвига с нелинейной обратной связью в поточном шифре Grain [3]. Бент-функции связаны также с некоторыми объектами теории кодирования, например с кодами Рида — Маллера [4].

Функция $g : \mathbb{Z}_4^n \to \mathbb{Z}_4$ называется *кватернарной функцией* от n переменных [5]. *Преобразование Уолша — Адамара кватернарной функции* g определяется следующим образом:

$$W_g(x) = \sum_{y \in \mathbb{Z}_4^n} i^{x.y+g(y)},$$

где «+» означает сложение по модулю 4.

[1]Работа выполнена в рамках государственного задания Института математики им. С. Л. Соболева СО РАН (проект № 0314-2019-0017) при поддержке РФФИ (проект № 18-07-01394) и Лаборатории криптографии JetBrains Research.

Кватернарная функция g от n переменных называется *бент-функцией*, если $|W_g(x)| = 4^{n/2}$ для любого $x \in \mathbb{Z}_4^n$.

Целью данной работы является изучение связи свойств «быть бент» кватернарных и булевых функций. Эта задача впервые поставлена в работе [6] (см. также [7]).

Каждая кватернарная функция g от n переменных может быть представлена для любых $x, y \in \mathbb{Z}_2^n$ следующим образом:

$$g(x + 2y) = a(x, y) + 2b(x, y).$$

Здесь сложение производится по модулю 4, а функции a и b — это *компонентные* булевы функции от $2n$ переменных.

Утверждение 1. Для любой кватернарной функции $g(x+2y) = a(x, y)+2b(x, y)$ от одной переменной, где $x, y \in \mathbb{Z}_2$, справедливо, что g — кватернарная бент-функция тогда и только тогда, когда $b(x, y)$ — бент-функция и $a(x, y)$ равна $0, 1, x$ или $x \oplus 1$. Кроме того, если g — кватернарная бент-функция, то b и $a \oplus b$ — булевы бент-функции.

Компьютерные вычисления показали, что количество кватернарных бент-функций от одной переменной равно 32.

Количество кватернарных бент-функций при $n = 2$ равно 200704. Среди них 98304 таких функций, что ни одна из булевых функций a, b и $a \oplus b$ не является бент-функцией, но при этом для 3072 из них a линейная. Существуют 36864 функции, таких, что b и $a \oplus b$ — бент-функции, при этом для 33792 из них функция a нелинейная, а для 2304 и 768 a является линейной функцией или константой соответственно. Количество кватернарных функций, для которых каждая из функций a, b и $a \oplus b$ — бент-функция, равно 16384. Для оставшихся 49152 функций a является бент-функцией, b и $a \oplus b$ — нелинейные булевы функции.

Теорема 1. Пусть $g(x + 2y) = a(x, y) + 2b(x, y)$ — кватернарная бент-функция, где $x, y \in \mathbb{Z}_2^n$; a, b — булевы функции от $2n$ переменных. Тогда b и $a \oplus b$ — нелинейные функции при любом числе переменных $n \geqslant 1$.

Следующие два утверждения показывают, что между свойствами «быть бент» кватернарной функции g и её компонентных булевых функций b и $a \oplus b$ нет прямой связи.

Утверждение 2. Для любого $n \geqslant 2$ существует кватернарная бент-функция $g(x+2y) = a(x, y)+2b(x, y)$ от n переменных, где b и $a \oplus b$ не являются бент-функциями от $2n$ переменных.

Утверждение 3. Для любого n существует кватернарная функция $g(x + 2y) = a(x, y) + 2b(x, y)$ от n переменных, которая не является бент-функцией, когда b и $a \oplus b$ — булевы бент-функции от $2n$ переменных.

Представим две простые конструкции для кватернарных бент-функций от любого числа переменных.

Утверждение 4. Кватернарная функция от n переменных

$$g(x_1 + 2x_{n+1}, \ldots, x_n + 2x_{2n}) = \sum_{i=1}^{n} 2x_i x_{i+n} + cx_j,$$

где $c \in \mathbb{Z}_2$, $j \in \{1, \ldots, n\}$ и «+» — сложение по модулю 4, является бент-функцией при любом n. Заметим, что при этом

$$b(x_1, \ldots, x_{2n}) = \bigoplus_{i=1}^{n} x_i x_{i+n} \quad \text{и} \quad a(x_1, \ldots, x_{2n}) \oplus b(x_1, \ldots, x_{2n}) = \bigoplus_{i=1}^{n} x_i x_{i+n} \oplus cx_j$$

— бент-функции от $2n$ переменных.

Утверждение 5. Пусть $g(x+2y) = a(x,y)+2b(x,y)$, где $x, y \in \mathbb{Z}_2^n$ и a, b — булевы функции от $2n$ переменных, является бент-функцией. Тогда функция $g'(x+2y) = 3a(x,y)+2b(x,y)$ также является кватернарной бент-функцией от $n \geqslant 1$ переменных.

Отметим, что утверждение верно и в обратную сторону.

ЛИТЕРАТУРА

1. *Matsui M.* Linear cryptanalysis method for DES cipher // Eurocrypt'1993. LNCS. 1994. V. 765. P. 386–397.
2. *Adams C.* Constructing symmetric ciphers using the CAST design procedure // Design, Codes, and Cryptography. 1997. V. 12. No. 3. P. 283–316.
3. *Hell M., Johansson T., Maximov A., and Meier W.* A stream cipher proposal: Grain-128 // IEEE Intern. Symp. Inform. Theory. Seattle, WA, 2006. P. 1614–1618.
4. *Tokareva N.* Bent Functions: Results and Applications to Cryptography. Acad. Press, Elsevier, 2015. 230 p.
5. *Kumar P. V., Scholtz R. A., and Welch L. R.* Generalized bent functions and their properties // J. Combin. Theory. 1985. V. 40. No. 1. P. 90–107.
6. *Solé P. and Tokareva N.* Connections Between Quaternary and Binary Bent Functions // Cryptology ePrint Archive, Report 2009/544. `http://eprint.iacr.org/`.
7. *Solé P. and Tokareva N.* On quaternary and binary bent functions // Прикладная дискретная математика. Приложение. 2009. № 1. С. 16–18.

UDC 519.7 DOI 10.17223/2226308X/13/11

ON A SECONDARY CONSTRUCTION OF QUADRATIC APN FUNCTIONS[1]

K. V. Kalgin, V. A. Idrisova

Almost perfect nonlinear functions possess the optimal resistance to the differential cryptanalysis and are widely studied. Most known constructions of APN functions are obtained as functions over finite fields $\mathbb{F}_{2^n}$ and very little is known about combinatorial constructions in $\mathbb{F}_2^n$. We consider how to obtain a quadratic APN function in $n+1$ variables from a given quadratic APN function in n variables using special restrictions on new terms.

Keywords: *vectorial Boolean function, APN function, quadratic function, secondary construction.*

Let us recall some definitions. Let $\mathbb{F}_2^n$ be the n-dimensional vector space over $\mathbb{F}_2$. A function F from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$, where n and m are integers, is called a *vectorial Boolean function*. If $m = 1$, such a function is called *Boolean*. Every vectorial Boolean function F can be represented as a set of m *coordinate functions* $F = (f_1, \ldots, f_m)$, where f_i is a Boolean function in n variables. Any vectorial function F can be represented uniquely in its *algebraic normal form* (ANF):

$$F(x) = \sum_{I \in \mathcal{P}(N)} a_I \Big(\prod_{i \in I} x_i\Big),$$

where $\mathcal{P}(N)$ is a power set of $N = \{1, \ldots, n\}$ and $a_I \in \mathbb{F}_2^m$. The *algebraic degree* of a given function F is the degree of its ANF: $\deg(F) = \max\{|I| : a_I \neq 0, I \in \mathcal{P}(N)\}$. If algebraic

[1]The work was carried out within the framework of the state contract of the Sobolev Institute of Mathematics (project no. 0314-2019-0017) and supported by RFBR (projects no. 18-07-01394, 20-31-70043) and Laboratory of Cryptography JetBrains Research.

degree of a function F is not more than 1, then F is called *affine*. If for an affine function F it holds $F(\mathbf{0}) = \mathbf{0}$, then F is called *linear*. If algebraic degree of a function F is equal to 2, then F is called *quadratic*. Two vectorial functions F and G are *extended affinely equivalent* (*EA-equivalent*) if $F = A_1 \circ G \circ A_2 + A$, where A_1, A_2 are affine permutations on $\mathbb{F}_2^n$ and A is an affine function. Let F be a vectorial Boolean function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$. For vectors $a, b \in \mathbb{F}_2^n$, where $a \neq 0$, consider the value

$$\delta(a, b) = \big|\{x \in \mathbb{F}_2^n : F(x + a) + F(x) = b\}\big|.$$

Denote by Δ_F the following value:

$$\Delta_F = \max_{a \neq \mathbf{0},\ b \in \mathbb{F}_2^n} \delta(a, b).$$

Then F is called *differentially* Δ_F*-uniform* function. The smaller the parameter Δ_F is, the better the resistance of a cipher containing F as an S-box to differential cryptanalysis. For the vectorial functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$, the minimal possible value of Δ_F is equal to 2. In this case, the function F is called *almost perfect nonlinear* (*APN*). This notion was introduced by K. Nyberg in [1]. APN fuctions draw attention of many researchers, but there is still a significant list (see, for example, [2 – 4]) of important open questions. We are especially interested how to find new constructions of APN functions in vector space $\mathbb{F}_2^n$, since almost all the known constructions of this class are found only as polynomials over the finite fields, and to the best of our knowledge, only a few approaches to such combinatorial constructions was proposed [5, 6].

Since EA-equivalence preserves APNness, it is always possible to omit linear and constant terms in the algebraic normal form of a given APN function. Further we will consider quadratic vectorial Boolean functions that have only quadratic terms in their ANF. The following theorem gives a necessary condition on the ANF of a given APN function.

Theorem 1 [7]. Let $F = (f_1, \ldots, f_n)$ be an APN function in n variables. Then every quadratic term $x_i x_j$, where $i \neq j$, appears at least in one coordinate function of F.

This property motivated us to suggest the following construction of quadratic APN functions. Let $G = (g_1, \ldots, g_n)$ be a quadratic APN-function in n variables. Consider vectorial function $F = (f_1, \ldots, f_n, f_{n+1})$ in $n + 1$ variables such that

$$\begin{aligned} f_1 &= g_1 + \sum_{i=1}^{n} \alpha_{1,i} x_i x_{n+1}, \\ &\ldots \\ f_n &= g_n + \sum_{i=1}^{n} \alpha_{n,i} x_i x_{n+1}, \\ f_{n+1} &= g_{n+1} + \sum_{i=1}^{n} \alpha_{n+1,i} x_i x_{n+1}, \end{aligned} \tag{1}$$

where $\alpha_{1,i} \ldots, \alpha_{n+1,i} \in \mathbb{F}_2$ for $i = 1, \ldots, n$ and $g_{n+1} = \sum\limits_{1 \leqslant j < k \leqslant n} \beta_{j,k} x_j x_k$ for some fixed $\beta_{j,k} \in \mathbb{F}_2$. Note that if $\alpha_{1,i}, \ldots, \alpha_{n,i}$ are such that each term $x_i x_{n+1}$ appears at least in one of the coordinate functions $f_1, \ldots, f_n$, then the necessary condition of Theorem 1 is held for the constructed function F.

Each quadratic vectorial function G in n variables can be considered as a symmetric matrix $\mathcal{G} = (g_{ij})$, where each element $g_{ij} \in \mathbb{F}_2^n$ is a vector of coefficients corresponding to

term $x_i x_j$ in the algebraic normal form of G and all diagonal elements g_{ii} are null. It is necessary to mention that these matrices are essentially the same as so-called QAM matrices that were used in [8, 9] to construct and classify a lot of new quadratic APN functions over finite fields. Using these matrices, the APN property can be formulated in the following way:

Proposition 1. Let $\mathcal{G}$ be the matrix that corresponds to quadratic vectorial function G. Then function G is APN if and only if $x\,(\mathcal{G} \cdot a) \neq 0$ for all $x \neq a$, where $a, x \in \mathbb{F}_2^n$ and $a \neq \mathbf{0}$.

In terms of matrices, the construction (1) can be considered as an extension of a given $\mathcal{G}$ with an extra bit that represents g_{n+1} in every element and an extra pair of row and column that represents a set of new terms $x_i x_{n+1}$.

Consider a quadratic APN function G and the corresponding $n \times n$ matrix $\mathcal{G}$. Denote the vector of nonzero coefficients as $\alpha = (\alpha_1, \ldots, \alpha_n)$. Let us fix g_{n+1} and construct $(n+1) \times$ $\times\, (n+1)$ matrix $\mathcal{F}$ by adding $(\alpha_1, \ldots, \alpha_n, 0)$ as the last column and the last row and adding new bit to every element according to the choice of g_{n+1}. Let us denote as $\mathcal{G}'$ the submatrix (f_{ij}) of $\mathcal{F}$, such that $i, j < n+1$. Let $\langle X \rangle$ denote the linear span of X and F be the quadratic vectorial function corresponding to the constructed matrix $\mathcal{F}$.

Theorem 2. A function F is APN if and only if $\alpha \cdot a'$ does not belong to $\langle \mathcal{G}' \cdot a' \rangle$ for all $a' \in \mathbb{F}_2^n$, $a' \neq \mathbf{0}$.

Theorem 2 shows how to choose new coefficients $\alpha_{1,i} \ldots, \alpha_{n+1,i} \in \mathbb{F}_2$ in the construction (1) such that an obtained function F is APN. When $n = 3, 4$ and 5, for APN functions that are representatives of EA classes, all possible classes of quadratic APNs are obtained for $4, 5$ and 6 variables from the classification [10] and large variety of classes for constructing functions in 6 and 7 variables.

REFERENCES

1. *Nyberg K.* Differentially uniform mappings for cryptography. EUROCRYPT'93, LNCS, 1994, vol. 765, pp. 55–64.
2. *Carlet C.* Open questions on nonlinearity and on APN Functions. WAIFI 2014, LNCS, 2015, vol. 9061, pp. 83–107.
3. *Glukhov M. M.* O priblizhenii diskretnykh funktsiy lineynymi funktsiyami [On the approximation of discrete functions by linear functions]. Matematicheskie Voprosy Kriptografii, 2016, vol. 7, no. 4, pp. 29–50. (in Russian)
4. *Tuzhilin M. E.* Pochti sovershennye nelineynye funktsii [APN-functions]. Prikladnaya Diskretnaya Matematika, 2009, no. 3(5), pp. 14–20. (in Russian)
5. *Gorodilova A. A.* Characterization of almost perfect nonlinear functions in terms of subfunctions. Discrete Math. Appl., 2016, vol. 26, iss. 4, pp. 193–202.
6. *Idrisova V. A.* On an algorithm generating 2-to-1 APN functions and its applications to "the big APN problem". Cryptogr. Commun., 2019, no. 11, pp. 21–39.
7. *Beth T. and Ding C.* On almost perfect nonlinear permutations. EUROCRYPT'93, LNCS, 1993, vol. 765, pp. 65–76.
8. *Yu Y., Wang M., and Li Y.* A matrix approach for constructing quadratic APN functions. Des. Codes Cryptogr., 2014, no. 73, pp. 587–600.
9. *Yu Y., Kaleyski N. S., Budaghyan L., and Li Y.* Classification of Quadratic APN Functions with Coefficients in GF(2) for Dimensions up to 9. IACR Cryptol. ePrint Arch.: 1491, 2019.
10. *Brinkmann M. and Leander G.* On the classification of APN functions up to dimension five. Des. Codes Cryptogr., 2008, vol. 49, iss. 1–3, pp. 273–288.

UDC 519.7 DOI 10.17223/2226308X/13/12

ON ONE-TO-ONE PROPERTY OF A VECTORIAL BOOLEAN FUNCTION OF THE SPECIAL TYPE[1]

M. M. Zapolskiy, N. N. Tokareva

S-boxes are widely used in cryptography. In particular, they form important components of SP and Feistel networks. Mathematically, S-box is a vectorial Boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ that should satisfy several cryptographic properties. Usually $n = m$. We study one-to-one property of a vectorial Boolean function constructed in a special way on the base of a Boolean function and a permutation on n elements. The number of all one-to-one functions of this type is calculated.

Keywords: *Boolean function, vectorial Boolean function, S-box.*

Let $\pi \in S_n$ be a permutation such that $\pi^n(x) = x$. Consider some $x \in \mathbb{F}_2^n$, $x = (x_1, \ldots, x_n)$, define $\pi(x) = (x_{\pi(1)}, \ldots, x_{\pi(n)})$. Let f be a Boolean function in n variables, we construct vectorial Boolean function $F_\pi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ by the following rule:

$$F_\pi(x) = (f(x), f(\pi(x)), f(\pi^2(x)), \ldots, f(\pi^{n-1}(x))).$$

Let $\Delta_{\pi,n}$ be the set of all these functions. Define $\rho(x) = (x_n, x_1, x_2, \ldots, x_{n-1})$, i.e., $\rho = (n, 1, 2, \ldots, n-1)$.

Proposition 1. Let $\pi \in S_n$ be such that $\pi^n(x) = x$, $F_\pi \in \Delta_{\pi,n}$. Then $F_\pi(\pi(x)) = \rho^{-1}(F_\pi(x))$ for all $x \in \mathbb{F}_2^n$.

We define action of π on $\mathbb{F}_2^n$ by the rule: if $x \in \mathbb{F}_2^n$, then $x \circ \pi = \pi(x)$. This action splits $\mathbb{F}_2^n$ into orbits with respect to π. If x is in some orbit o, we call x *a generator of* o. We call $O_\pi(x)$ *the orbit with respect to the action of* π.

Example 1. For $n = 4$, the set $\mathbb{F}_2^n$ is divided into six orbits with respect to the permutation ρ:

$O_\rho((0,0,0,0))$	$(0,0,0,0)$
$O_\rho((1,0,0,0))$	$(1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1)$
$O_\rho((1,0,1,0))$	$(1,0,1,0), (0,1,0,1)$
$O_\rho((1,0,0,1))$	$(1,0,0,1), (1,1,0,0), (0,1,1,0), (0,0,1,1)$
$O_\rho((0,1,1,1))$	$(0,1,1,1), (1,0,1,1), (1,1,0,1), (1,1,1,0)$
$O_\rho((1,1,1,1))$	$(1,1,1,1)$

We denote by $\Theta_{\pi,n}$ the set of all orbits with respect to the action of π on $\mathbb{F}_2^n$. Proposition 1 implies that, for arbitrary $F_\pi \in \Delta_{\pi,n}$, values of elements of some π-orbit $g \in \Theta_{\pi,n}$ are elements of some ρ-orbit $q \in \Theta_{\rho,n}$, since $F_\pi(\pi^i(x)) = \rho^{-i}(F_\pi(x))$. Let $M_{\pi,n}^k = \{g \in \Theta_{\pi,n} : |g| = k\}$.

Let $\Psi_{F_\pi,n} : \Theta_{\pi,n} \to \Theta_{\rho,n}$ be a mapping defined by the rule $\Psi_{F_\pi,n}(O_\pi(x)) = O_\rho(F_\pi(x))$. Now we can formulate conditions for F_π to be one-to-one in terms of $\Psi_{F_\pi,n}$.

Theorem 1. $F_\pi \in \Delta_{\pi,n}$ is an one-to-one function if and only if $\Psi_{F_\pi,n}$ is one-to-one. If $\Psi_{F_\pi,n}$ is one-to-one, then $|\Psi_{F_\pi,n}(g)| = |g|$, for all $g \in \Theta_{\pi,n}$.

As a corollary of Theorem 1, we obtain the following result.

[1]The work was carried out within the framework of the state contract of the Sobolev Institute of Mathematics (project no. 0314-2019-0017) and supported by RFBR (project no. 18-07-01394) and Laboratory of Cryptography JetBrains Research.

Proposition 2. If $|M^k_{\pi,n}| \neq |M^k_{\rho,n}|$ for some k, then the set of one-to-one functions from $\Delta_{\pi,n}$ is empty.

Theorem 1 means that in order to construct one-to-one functions $F_\pi \in \Delta_{\pi,n}$ we can use bijective maps $\Psi_n : \Theta_{\pi,n} \to \Theta_{\rho,n}$ that satisfy $|\Psi_n(g)| = |g|$, where $g \in \Theta_{\pi,n}$. Then, depending on them, we can construct $F_\pi \in \Delta_{\pi,n}$ such that $\Psi_{F_\pi,n} \equiv \Psi_n$.

Proposition 3. Let $\Psi_n : \Theta_{\pi,n} \to \Theta_{\rho,n}$ satisfy $|\Psi_n(g)| = |g|$ for all $g \in \Theta_{\pi,n}$. Then, for all $k \in \mathbb{N}$, the restriction of Ψ_n on $M^k_{\pi,n}$ is a permutation of $M^k_{\pi,n}$.

Now consider the case $\pi = \rho$. We define $M^k_n = M^k_{\rho,n}$. Consider an one-to-one function Ψ_n which satisfies $|\Psi_n(g)| = |g|$ for all $g \in \Theta_{\pi,n}$. Let us construct function $F_\rho \in \Delta_{\rho,n}$ based on Ψ_n. Let $O \in \Theta_{\rho,n}$ be an orbit of length k. If the value of F_ρ for some $x \in O$ is determined, then the value of F_ρ is determined for all $x \in O$, since $F_\rho(\rho^n(x)) = \rho^{-n}(F_\rho(x))$. Thus, for every $\Psi_{F_\rho,n}$, we are able to construct $\prod\limits_{k \in I_n} k^{|M^k_n|}$ functions, where $I_n = \{z \in \mathbb{N} : z|n\}$, and all of them are pairwise different.

Proposition 4. For any $k \in \mathbb{N}$, $\sum\limits_{\ell \in I_k} \ell \cdot |M^\ell_n| = 2^k$.

This formula allows us to calculate $|M^k_n|$ for every k. There are always only two orbits of length one, so we can calculate $|M^k_n|$ for every prime k. Then we can calculate it for every k. Therefore, we get the number of one-to-one functions from $\Delta_{\rho,n}$:

Theorem 2. The number of one-to-one vectorial Boolean functions in class $\Delta_{\rho,n}$ is equal to $\prod\limits_{k \in I_n} |M^k_n|! \cdot k^{|M^k_n|}$.

UDC 519.7 DOI 10.17223/2226308X/13/13

CRYPTOGRAPHIC PROPERTIES OF A SIMPLE S-BOX CONSTRUCTION BASED ON A BOOLEAN FUNCTION AND A PERMUTATION[1]

D. A. Zyubina, N. N. Tokareva

We propose a simple method of constructing S-boxes using Boolean functions and permutations. Let π be an arbitrary permutation on n elements, f be a Boolean function in n variables. Define a vectorial Boolean function $F_\pi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ as $F_\pi(x) = (f(x), f(\pi(x)), f(\pi^2(x)), \ldots, f(\pi^{n-1}(x)))$. We study cryptographic properties of F_π such as high nonlinearity, balancedness, low differential δ-uniformity in dependence on properties of f and π for small n.

Keywords: *Boolean function, vectorial Boolean function, S-box, high nonlinearity, balancedness, low differential δ-uniformity, high algebraic degree.*

S-boxes play the crucial role for providing resistance of a block cipher to different types of attacks. The major reason for this is that in classical and modern block ciphers the main complicated and nonlinear layer is presented namely by S-boxes. Mathematically, S-box is a vectorial Boolean function that maps n bits to m bits. Usually, n coincides with m. It is well known that some special mathematical properties of S-boxes, such as high nonlinearity, low differential uniformity, high algebraic immunity, etc. make a

[1]The work is supported by Mathematical Center in Akademgorodok under agreement No. 075-15-2019-1613 with the Ministry of Science and Higher Education of the Russian Federation and Laboratory of Cryptography JetBrains Research.

cipher with such S-boxes be resistant to linear, differential, algebraic and other methods of cryptanalysis. The cryptographic properties of a Boolean (vectorial) function contradict to each other [1, 2]. That is why we try to find vectorial Boolean functions that reach a tradeoff between different cryptographic properties and are constructed using mathematical methods (and not a direct computer search) for their constructing.

In the paper, we propose a simple method of constructing S-boxes using Boolean functions. Let π be an arbitrary permutation on n elements, $\pi \in \mathbb{S}_n$. If $x = (x_1, \ldots, x_n)$ is a binary vector, then let $\pi(x)$ be a vector $\pi(x) = (x_{\pi(1)}, \ldots, x_{\pi(n)})$. Let f be a Boolean function in n variables. Define a vectorial Boolean function $F_\pi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ as follows:

$$F_\pi(x) = (f(x), f(\pi(x)), f(\pi^2(x)), \ldots, f(\pi^{n-1}(x))).$$

We would like to study cryptographic properties of the vectorial Boolean function F_π in dependence on properties of the Boolean function f and the permutation π.

Note that this way of constructing vectorial Boolean functions was already mentioned before but only for obtaining some examples. Thus, A. Udovenko proposed a vectorial Boolean function of this type in 5 variables with the maximal possible algebraic immunity 3. It is a unique known solution of the previously unsolved problem from NSUCRYPTO 2016 [3]. So functions F_π can have good crypto properties.

Separately, we consider the special case of a permutation. Let A_n be the set of all full cycle permutations for n elements. For example, A_4 consists of 6 permutations: $(2, 3, 4, 1)$, $(2, 4, 1, 3)$, $(3, 1, 4, 2)$, $(3, 4, 2, 1)$, $(4, 1, 2, 3)$, $(4, 3, 1, 2)$ presented as vectors or (1234), (1243), (1342), (1324), (1432), (1423) in cyclic representation.

Let us recall definitions of several cryptographic properties.

A Boolean function f in n variables is called *balanced* if it takes every value (0 or 1) the same number of times [4]. A vectorial Boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is *balanced* if it takes every value of $\mathbb{F}_2^n$ equally often [2] .

Let $\mathcal{A}_n = \{\langle a, x\rangle \oplus b : a \in \mathbb{F}_2^n, b \in \mathbb{F}_2\}$ be the class of all affine Boolean functions in n variables [5]. The *nonlinearity* $\mathrm{nl}(f)$ of a Boolean function f in n variables is the Hamming distance between f and the set of all affine Boolean functions in n variables [5]. The *nonlinearity* $\mathrm{nl}(F)$ of a vectorial Boolean function F is the minimal nonlinearity of all its component Boolean functions:

$$\mathrm{nl}(F) = \min_{v \in \mathbb{F}_2^n \setminus \{0\}} \mathrm{nl}(F_v) = \min_{v \in \mathbb{F}_2^n \setminus \{0\}} \mathrm{d}(\langle v, F\rangle, \mathcal{A}_n) = \min_{v \in \mathbb{F}_2^n \setminus \{0\}} \min_{g \in \mathcal{A}_n} \mathrm{d}(\langle v, F\rangle, g).$$

The *algebraic degree* of a vectorial Boolean function is the maximal algebraic degree of its component functions [2]. Note that for our construction $\deg(F) = \deg(f)$ for any π, since all coordinate functions of F have degree $\deg(f)$.

For a vectorial Boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ let δ_F denote the maximal number of solutions for the equation $F(x) \oplus F(x \oplus a) = b$ while a, b run through $\mathbb{F}_2^n$ and a is nonzero. Then F is called *differential δ_F-uniform* [2]. Note that the minimal possible value of δ_F, where F maps from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$, is 2.

We consider cryptographic properties of F_π for small n in relation to f and π. All of the following propositions are obtained via computer search.

1. Case $n = 2$

• For any permutation $\pi \in \mathbb{S}_2$ there exists a Boolean function f in 2 variables such that $\delta_{F_\pi} = 2$. Moreover, such Boolean functions are constructed as $f(x) = x_1x_2 \oplus a_1x_1 \oplus a_2x_2 \oplus a_0$, where $a_0, a_1, a_2 \in \mathbb{F}_2$.

2. Case $n = 3$

For any Boolean function f in 3 variables $\mathrm{nl}(f) \leqslant 2$.

• For any permutation $\pi \in A_3$ there exists a balanced Boolean function f in 3 variables such that vectorial Boolean function F_π is balanced.

• For any permutation $\pi \in A_3$ it holds $\mathrm{nl}(F_\pi) = \mathrm{nl}(f)$. Note that if $\mathrm{nl}(F_\pi) = 2$, i.e., is maximal, then $\delta_{F_\pi} = 2$, i.e., is minimal possible. The number of such funstions f is 48.

• For an arbitary permutation $\pi \notin A_3$ and Boolean function f in 3 variables $\delta_{F_\pi} \geqslant 4$.

3. Case $n = 4$

Let us introduce the notation for permutations from the set A_4: $\pi_1 = (2, 3, 4, 1)$, $\pi_2 = = (4, 1, 2, 3)$, $\pi_3 = (2, 4, 1, 3)$, $\pi_4 = (3, 1, 4, 2)$, $\pi_5 = (3, 4, 2, 1)$, $\pi_6 = (4, 3, 1, 2)$. Note that $\pi_1^{-1} = \pi_2$, $\pi_3^{-1} = \pi_4$, $\pi_5^{-1} = \pi_6$.

• For any permutation $\pi \in A_4^1$ and a balanced Boolean function f in 4 variables such that $\delta_{F_\pi} = 2$, F_π is not balanced.

• For any permutation $\pi \in A_4^1$ there exists a Boolean function f in 4 variables such that if $\delta_{F_\pi} = 2$ and nonlinearity of f and F_π are the same, then $\delta_{F_{\pi^{-1}}} = 2$. Moreover, nonlinearity of $F_{\pi^{-1}}$ and f coincide.

• For any permutation $\pi \notin A_4^1$ for an arbitary Boolean function f in 4 variables $\delta_{F_\pi} \geqslant 4$.

Based on the results, we suppose that it is possible to construct vectorial Boolean functions in the arbitrary number of variables with cryptographic properties good enough using our simple construction for necessary Booleans functions and permutations.

We plan to use our program for studying vectorial Boolean functions with larger number of variables, now this work is in progress.

REFERENCES

1. *Cusick T. W. and Stănică P.* Cryptographic Boolean Functions and Applications. USA, Acad. Press, Elsevier, 2009.
2. *Carlet C.* Vectorial Boolean functions for cryptography. Y. Crama and P. Hammer (eds.) Boolean Models and Methods in Mathematics, Computer Science, and Engineering. Cambridge, Cambridge University Press, 2010, pp. 398–470.
3. *Tokareva N., Gorodilova A., Agievich S., et al.* Mathematical methods in solutions of the problems presented at the Third International Students' Olympiad in Cryptography. Prikladnaya Diskretnaya Matematika, 2018, no. 40, pp. 34–58.
4. *Carlet C.* Boolean functions for cryptography and error-correcting codes. Y. Crama and P. Hammer (eds.) Boolean Models and Methods in Mathematics, Computer Science, and Engineering. Cambridge, Cambridge University Press, 2010, pp. 257–397.
5. *Logachev O. A., Salnikov A. A., Smyshlyaev S. V., and Yaschenko V. V.* Bulevy funktsii v teorii kodirovaniya i kriptologii [Boolean Functions in Coding Theory and Cryptology]. Moscow, MCCME Publ., 2012. 584 p. (in Russian)

Секция 3

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 511.48 DOI 10.17223/2226308X/13/14

ПРОЕКТ СТАНДАРТИЗАЦИИ ПОСТКВАНТОВОЙ ЦИФРОВОЙ ПОДПИСИ

Е. А. Киршанова, Н. С. Колесников, Е. С. Малыгина, С. А. Новоселов

Предлагается цифровая подпись, безопасность которой основана на задачах MLWR и MSIS в алгебраических решётках. Конструкция подписи основана на парадигме Фиата — Шамира. Доказывается безопасность схемы в квантовой модели безопасности и описываются конкретные параметры, при которых схема достигает уровня безопасности в 100 бит. Благодаря модульной структуре решёток, уровень безопасности легко изменить в большую или меньшую стороны. Наше предложение может служить основой проекта по стандартизации постквантовых примитивов на решётках.

Ключевые слова: *цифровая подпись, криптография на решётках, постквантовая криптография, парадигма Фиата — Шамира.*

Введение

Криптографические примитивы на решётках — одно из самых обещающих направлений современной криптографии не только ввиду стойкости этих примитивов к атакам на квантовом компьютере, но и вследствие большого спектра конструкций (гомоморфное шифрование, электронные голосования, различные типы подписей), а также их надёжности по отношению к *классическим* атакам. Криптографические конструкции на решётках не только элегантны в теории, но и значимы на практике, поэтому в достаточно скором будущем будут стандартизированы. Пробные версии обмена ключами New Hope уже тестированы в TLS-соединениях для браузера Google Chrome [1]. Процесс стандартизации постквантовых схем доступен по адресу `https://csrc.nist.gov/projects/post-quantum-cryptography`.

В этой работе мы предлагаем схему цифровой подписи, основанную на алгебраических решётках, конструкция которой удовлетворяет следующими основным свойствам:

1) безопасность схемы основана на задачах «в среднем», а именно на задачах LWR (Learning With Rounding) и SIS (Shortest Integer Solution) — классических трудных задачах на решётках, определения которых даны в п. 1);
2) для эффективности схемы используется так называемый *модульный* вариант задач, а именно module-LWR, module-SIS [2], что не только позволяет уменьшить размеры параметров схемы и время операций, но и даёт возможность легко варьировать уровни безопасности схемы;
3) стойкость схемы доказана в квантовой модели QROM (Quantum Random Oracle Model) для «сильного» атакующего, а именно для атаки вида $\mathsf{UF}-\mathsf{sCMA}$; доказательство можно найти в [3];

4) в процессе генерации ключей и подписи вместо нормального распределения используется равномерное распределение из интервала, что уменьшает риск сторонних атак;
5) предлагается конкретный набор параметров схемы с битовой оценкой сложности атак на предложенные параметры (см. п. 3).

Представленная здесь схема основана на парадигме Фиата — Шамира [4, 5] и по идеологии продолжает серию работ, предлагающих конкретные схемы подписи [6 – 8]. Основное отличие нашей схемы от ранее предложенных заключается в том, что безопасность ключей основана на задаче LWR (а не на задаче LWE (Learning With Errors)). Мы считаем, что такой подход упрощает описание и потенциально ускоряет вычисления.

1. Предварительные сведения

1.1. Обозначения

Будем обозначать $\mathbb{Z}/q\mathbb{Z}$ кольцо целых по чётному модулю q, результат $z \bmod q$ представляем в интервале $\{0, \ldots, q-1\}$; R, R_q и R_p — кольца многочленов $\mathbb{Z}[x]/(x^n+1)$, $\mathbb{Z}/q\mathbb{Z}[x]/(x^n+1)$ и $\mathbb{Z}/p\mathbb{Z}[x]/(x^n+1)$ соответственно. Векторы будем обозначать жирными строчными буквами (например, $\mathbf{x}$), матрицы — прописными (например, $\mathbf{A}$), константы — обычными строчными; $\mathbb{I}$ — единичная матрица. Элементы кольца $\mathbb{Z}[x]/(x^n+1)$ будем понимать как векторы-коэффициенты многочленов. Векторы по умолчанию являются вектор-столбцами. Евклидова (или ℓ_2) норма вектора $\mathbf{x}$ определяется как $\|\mathbf{x}\| = \|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$, а ℓ_∞-норма — как $\|\mathbf{x}\|_\infty = \max_i |x_i|$.

Многочленам из кольца R ставим в соответствие векторы-коэффициенты длины n, поэтому произведение векторов $\mathbf{x} \cdot \mathbf{y}$ надо понимать как произведение соответствующих многочленов. Элементу $\mathbf{a} \in R_q$ ставим в соответствие матрицу $\mathrm{rot}(\mathbf{a}) \in (\mathbb{Z}/q\mathbb{Z})^{n\times n}$, i-я строка которой — коэффициенты многочлена $x^{i-1} \cdot \mathbf{a}$. Такая матрица задаёт произведение любого элемента из R_q на многочлен $\mathbf{a}$.

Для конечного множества S запись $s \leftarrow S$ обозначает, что s выбрано в соответствии со случайным равномерным распределением на S. Через S_β^ℓ обозначим множество векторов длины ℓ, каждый коэффициент которого взят в соответствии с равномерным распределением из множества $\{-\beta, \ldots, \beta\}$.

Для любого $x \in \mathbb{Q}$ запись $\mathsf{Round}(x) \in \mathbb{Z}$ означает взятие ближайшего целого, где $1/2$ округляется до 1. Для целого x функция $\mathsf{MSB}(x, d)$ (соответственно $\mathsf{LSB}(x, d)$) означает взятие d старших (соответственно младших) бит. Все операции распространяются на векторы и матрицы покоэффициентно.

В нашей схеме мы будет использовать два модуля: $q = 2^\nu$ и $p = 2^\mu$. «Конвертирование» элемента $x \in \mathbb{Z}/q\mathbb{Z}$ в $x' \in \mathbb{Z}/p\mathbb{Z}$ происходит по правилу $x' = \mathsf{Round}(x \cdot p/q)$. Так как модули — степени двойки, этот же результат можно получить, добавив к x константу $h = 2^{\nu-\mu-1}$ и взяв μ старших бит: $x' = \mathsf{MSB}(x + h, \mu)$. Такое представление операции Round использовано, например, в [9]. Вектор, каждая координата которого равна h, обозначим $\mathbf{h}$. Для всякого целого $w > 0$ положим $B_w = \{\mathbf{x} \in R : \|\mathbf{x}\|_\infty = 1, \|\mathbf{x}\| = \sqrt{w}\} \subseteq R$.

1.2. Синтаксис и модели безопасности цифровых подписей

Определение 1. Цифровая подпись — примитив, состоящий из трёх алгоритмов:

— вероятностный алгоритм генерации ключевой пары $\mathsf{KeyGen}(\mathsf{par})$, возвращающий секретный ключ sk и ключ верификации vk;

— вероятностный алгоритм генерации подписи $\mathsf{Sign}(\mathsf{sk}, m)$, который для сообщения $m \in \mathcal{M}$ возвращает подпись σ;
— детерминированный алгоритм $\mathsf{Verify}(m, \sigma, \mathsf{vk})$, который возвращает либо «Accept» (подпись σ корректна для (m, vk)), либо «Reject» (подпись σ не корректна для (m, vk)).

Цифровая подпись корректна с долей ошибки ε, если для всех пар $(\mathsf{sk}, \mathsf{vk}) \in \in \mathsf{KeyGen}(\mathsf{par})$ и всех сообщений $m \in \mathcal{M}$ имеем

$$\mathsf{P}\left[\mathsf{Verify}(m, \mathsf{Sign}(\mathsf{sk}, m), \mathsf{vk}) = \text{«Accept»}\right] \geqslant 1 - \varepsilon.$$

1.3. Сложные задачи на решётках

Безопасность нашей подписи основывается на двух «сложных в среднем» задачах. Первая — задача Обучения с Округлением (Learning With Rounding (LWR)) [10] — детерминированная версия задачи Обучения с Ошибками (Learning With Errors (LWE)) [11]. В основе безопасности ключей подписи лежит трудность этой модульной версии задачи над фактор-кольцом R_q [2]. Все вычисления производятся в фактор-кольце R_q, матрица $\mathbf{A}$ формируется как блочная матрица из $k \cdot \ell$ элементов из R_q, где каждый блок — матрица $\mathrm{rot}(\mathbf{a})$.

Для предлагаемой схемы, в отличие от классических задач LWR и LWE, где матрица $\mathbf{A}$ берётся случайным образом из $R_q^{k \times \ell}$, будем требовать, чтобы хотя бы один из $k \cdot \ell$ многочленов был обратим в R_q. Будем обозначать такую матрицу через $\widetilde{\mathbf{A}}$. Это требование не влияет на безопасность схемы, поскольку, как минимум, константное число многочленов обратимы в R_q[1]. Значит, если атакующий имеет непренебрежимо малую вероятность успеха для $\widetilde{\mathbf{A}}$, этот же атакующий имеет непренебрежимо малую вероятность успеха для $\mathbf{A} \leftarrow R_q$.

Определение 2 (*задача обучения с округлением* (*MLWR*)). Пусть $q \geqslant p \geqslant 1$, $k, \ell \geqslant 1$ — целые числа. MLWR-распределение для вектора $\mathbf{s} \leftarrow R_q^{\ell}$ есть множество пар вида $\left(\mathbf{A}, \mathsf{Round}\left(\frac{p}{q} \cdot \mathbf{A} \cdot \mathbf{s}\right)\right)$, где $\widetilde{\mathbf{A}} \leftarrow R_q^{k \times \ell}$. *Задача поиска*: для заданного произвольным образом большого числа выборок из MLWR-распределения для вектора $\mathbf{s} \leftarrow R_q^{\ell}$ восстановить $\mathbf{s}$. *Задача различения распределений*: для заданного произвольным образом большого числа выборок из $\widetilde{R}_q^{k \times \ell} \times R_p^k$ определить, являются ли они равномерно распределёнными или MLWR-распределёнными для вектора $\mathbf{s} \leftarrow R_q^{\ell}$.

Обе версии задачи эквивалентны (то есть, имея оракул, решающий одну задачу, можно решить другую за полиномиальное от n время) [12]. В доказательстве безопасности схемы подписи нам понадобится вторая версия. Безопасность подписи основана на задаче нахождения Короткого Целочисленного Решения (Short Integer Solution (SIS) problem) [13]. Нам потребуется модульная версия этой задачи.

Определение 3 (*задача нахождения Короткого Целочисленного Решения* (MSIS)). Зафиксируем $b \in \mathbb{N}$ и пусть $\mathbf{A} \leftarrow R_q^{k \times \ell}$. Модульная задача нахождения короткого целочисленного решения, параметризованная посредством $b > 0$, заключается в нахождении «короткого» ненулевого прообраза $\mathbf{y} \leftarrow R_q^{k+\ell}$ в решётке, определяемой $\mathbf{A}$, т. е.

$$\mathbf{y} \neq 0, \quad [\mathbb{I} | \mathbf{A}] \cdot \mathbf{y} = 0 \quad \text{и} \quad \|\mathbf{y}\|_{\infty} \leqslant b.$$

[1] Вероятность обратимости случайного многочлена в R_q, где q — степень двойки, не столь тривиальна (и не столь велика), как в случае простого q. Случайный многочлен $\mathbf{a} \in R_q$ обратим тогда и только тогда, когда $\mathrm{rot}(\mathbf{a})$ — обратимая матрица в $\mathbb{Z}/q\mathbb{Z}^{n \times n}$, что, в свою очередь, верно тогда и только тогда, когда $\det(\mathrm{rot}(\mathbf{a}))$ — обратимый элемент в $\mathbb{Z}/q\mathbb{Z}$. В случае $q = 2^{\nu}$ имеем $|\mathbb{Z}_q^*| = 2^{\nu-1}$, а значит, случайный элемент из $\mathbb{Z}/q\mathbb{Z}$ обратим с вероятностью $|\mathbb{Z}_q^*|/q = 1/2$.

Для доказательства безопасности схемы потребуется вариант задачи SIS, так называемый SelfTargetSIS, предложенный в [14]. В этой же работе описана редукция от SIS к SelfTargetSIS.

Определение 4 (*задача* SelfTargetSIS). Пусть $\mathcal{H} : \{0,1\}^* \to B_w$ — криптографическая хэш-функция. Зададим случайным образом $\mathbf{A} \leftarrow R_q^{k\times\ell}$ и доступ к квантовому случайному оракулу $\mathcal{H}(\cdot)$. Для исходного сообщения $M \in \{0,1\}^*$ задача SelfTargetSIS сводится к нахождению

$$\mathbf{y} = [\mathbf{r}, \mathbf{c}]^T, \quad \text{где} \quad 0 \leqslant \|\mathbf{y}\|_\infty \leqslant \gamma, \quad \mathcal{H}([\mathbf{A}|\mathbb{I}] \cdot \mathbf{y}, M) = \mathbf{c}.$$

2. Описание схемы

Цифровая подпись (алгоритмы 1–3) зависит от следующих параметров: $q = 2^\nu$, $p = 2^\mu$, $\nu > \mu$. Используется криптографическая хэш-функция $\mathcal{H} : \{0,1\}^* \to B_w$ [7]. Параметры k, ℓ отвечают за размерности ключей; s, γ задают интервалы для коэффициентов многочленов в процессе генерации ключей или подписи; d, β отвечают за корректность и безопасность схемы. Подпись формируется для сообщений $M \in \{0,1\}^*$. Конкретные значения параметров заданы в п. 3.

Алгоритм 1. Генерация ключей

Вход: $\ell > k > 1$, $q > p$, s.
Выход: $\mathbf{A}$, $\mathbf{t}$.
1: $\mathbf{A} \leftarrow R_q^{k\times\ell}$;
2: $\mathbf{s} \leftarrow S_s^\ell$;
3: $\mathbf{t} := \mathsf{Round}\left(\frac{p}{q} \cdot \mathbf{As}\right)$. // $\|\mathbf{t} - \mathbf{As}\|_\infty \leqslant 2^{\nu-\mu}$
4: **Вернуть** $\mathsf{sk} = \mathbf{s}$, $\mathsf{vk} = (\mathbf{A}, \mathbf{t})$.

Алгоритм 2. Генерация подписи

Вход: $q = 2^\nu$, $p = 2^\mu$, $\ell > 1$, M, $\mathbf{A}$, $\mathbf{t}$, $\mathbf{s}$, d, $\mathcal{H}$, β, γ, w.
Выход: $(\mathbf{z}, \mathbf{c})$.
1: $\mathbf{y} \leftarrow S_{\gamma-1}^\ell$;
2: $\mathbf{c} := \mathcal{H}\left(\mathsf{MSB}(\mathbf{A} \cdot \mathbf{y}, d), M\right)$;
3: $\mathbf{z} := \mathbf{y} + \mathbf{sc}$;
4: $\mathbf{w} := \mathbf{Az} - \mathbf{t} \cdot 2^{\nu-\mu} \cdot \mathbf{c}$;
5: **Если** $(\|\mathsf{LSB}(\mathbf{w}, \nu - d)\|_\infty \geqslant 2^{\nu-d} - w \cdot 2^{\nu-\mu+1})$ или $(\|\mathbf{z}\|_\infty \geqslant \gamma - \beta)$, **то** **restart**.
6: **Вернуть** $(\mathbf{z}, \mathbf{c})$.

Алгоритм 3. Проверка подписи

Вход: M, $\mathbf{z}$, $\mathbf{c}$, $\mathbf{A}$, $\mathbf{t}$, d, $\mathcal{H}$, β, γ.
Выход: «Accept» или «Reject».
1: $\mathbf{w} := \mathbf{A}\mathbf{z} - \mathbf{t} \cdot 2^{\nu-\mu} \cdot \mathbf{c}$;
2: $\mathbf{c}' := \mathcal{H}(\mathsf{MSB}(\mathbf{w}, d)), M)$;
3: **Если** $\mathbf{c}' = \mathbf{c}$ и $\|\mathbf{z}\|_\infty \leqslant \gamma - \beta$, **то**
4: **Вернуть** «Accept»,
5: **иначе**
6: **Вернуть** «Reject».

2.1. Корректность

Поскольку $\mathbf{w} = \mathbf{A} \cdot \mathbf{z} - \mathbf{t} \cdot 2^{\nu-\mu} \cdot \mathbf{c}$, $\mathbf{z} = \mathbf{y} + \mathbf{s} \cdot \mathbf{c}$ и $\mathbf{t} = \mathsf{Round}\left(\frac{p}{q} \cdot \mathbf{A}\mathbf{s}\right)$, то

$$\mathbf{w} = \mathbf{A} \cdot (\mathbf{y} + \mathbf{s} \cdot \mathbf{c}) - \mathbf{c} \cdot 2^{\nu-\mu} \cdot \mathsf{Round}\left(\frac{p}{q} \cdot \mathbf{A}\mathbf{s}\right) = \mathbf{A}\mathbf{y} + \mathbf{A}\mathbf{s}\mathbf{c} - \mathbf{c} \cdot 2^{\nu-\mu} \cdot \mathsf{Round}\left(\frac{p}{q} \cdot \mathbf{A}\mathbf{s}\right).$$

Согласно введённым обозначениям, $\mathsf{Round}\left(\frac{p}{q} \cdot \mathbf{A}\mathbf{s}\right) = \mathsf{MSB}(\mathbf{A}\mathbf{s} + \mathbf{h}, \mu)$, где $\mathbf{h}$ — вектор, каждая координата которого равна $h = 2^{\nu-\mu-1}$. Тогда

$$\mathbf{w} = \mathbf{A}\mathbf{y} + \mathbf{A}\mathbf{s}\mathbf{c} - \mathbf{c} \cdot 2^{\nu-\mu} \cdot \mathsf{MSB}(\mathbf{A}\mathbf{s} + \mathbf{h}, \mu) = \mathbf{A}\mathbf{y} + \mathbf{A}\mathbf{s}\mathbf{c} - \mathbf{c}\left(\mathbf{A}\mathbf{s} + \mathbf{h} + \mathsf{LSB}(\mathbf{A}\mathbf{s} + \mathbf{h}, \nu - \mu)\right).$$

Раскрывая скобки, окончательно получаем

$$\mathbf{w} = \mathbf{A}\mathbf{y} - \mathbf{c}\left(\mathbf{h} + \mathsf{LSB}(\mathbf{A}\mathbf{s} + \mathbf{h}, \nu - \mu)\right), \tag{1}$$

где $\|\mathbf{c}\left(\mathbf{h} + \mathsf{LSB}(\mathbf{A}\mathbf{s} + \mathbf{h}, \nu - \mu)\right)\|_\infty < w \cdot 2^{\nu-\mu+1}$, поскольку $\mathbf{c} \in B_w$ и $\|\mathsf{LSB}(\mathbf{A}\mathbf{s}, \nu - \mu)\|_\infty < 2^{\nu-\mu}$. Рассматривая $\mathsf{LSB}(\mathbf{w}, \nu - d)$ в алгоритме 2 на шаге 5 и учитывая ошибку $\mathbf{c}\left(\mathbf{h} + \mathsf{LSB}(\mathbf{A}\mathbf{s} + \mathbf{h}, \nu - \mu)\right)$, получаем, что при $\|\mathsf{LSB}(\mathbf{w}, \nu - d)\|_\infty > 2^{\nu-d} - w \cdot 2^{\nu-\mu+1}$ алгоритм отклоняет значение $\mathbf{w}$.

Так как $\mathbf{c}\left(\mathbf{h} + \mathsf{LSB}(\mathbf{A}\mathbf{s} + \mathbf{h}, \nu - \mu)\right)$ — малый вектор ошибки, то из равенства (1) очевидно, что $\mathsf{MSB}(\mathbf{w}, d) = \mathsf{MSB}(\mathbf{A}\mathbf{y}, d)$. Следовательно, вычисление $\mathbf{c}'$ на шаге 2 алгоритма 3 совпадает со значением вектора $\mathbf{c}$ на шаге 2 алгоритма 2.

2.2. О числе итераций в процедуре Sign

В процессе вычисления подписи алгоритм 2 на шаге 5 проверяет, попадают ли коэффициенты вектора $\mathbf{z}$ в интервал $\{-(\gamma - \beta - 1), \ldots, \gamma - \beta - 1\}$. Для фиксированного ключа $\mathbf{s}$ вероятность этого события зависит от $\|\mathbf{y}\|_\infty$, выбранного на шаге 1. Вычислим эту вероятность.

Пусть $\mathbf{z} = \mathbf{y} + \mathbf{v}$ такой, что $\mathbf{z} \in S^\ell_{\gamma-\beta-1}$. Обозначим $\beta = \|\mathbf{c}\mathbf{s}\|_\infty$. Так как $\|\mathbf{s}\|_\infty \leqslant s$ и $\mathbf{c} \in B_w$, то $\beta < ws$. Отсюда $\|\mathbf{v}\|_\infty \leqslant \beta$. Для каждого коэффициента $\mathbf{v}_i$ вектора $\mathbf{v}$ соответствующий коэффициент $\mathbf{z}_i$ лежит в интервале $\{-(\gamma - \beta - 1), \ldots, \gamma - \beta - 1\}$. Поскольку $\mathbf{y} = \mathbf{z} - \mathbf{v}$, то $\mathbf{y} \in S^\ell_{\gamma-1}$ и соответствующий коэффициент $\mathbf{y}_i$ лежит в интервале $\{-(\gamma - 1), \ldots, \gamma - 1\}$. Следовательно,

$$p_1 = \mathsf{P}_{\mathbf{y} \leftarrow S^\ell_{\gamma-1}}\left[\|\mathbf{z}\|_\infty < \gamma - \beta\right] = \frac{\left|S^\ell_{\gamma-\beta-1}\right|}{\left|S^\ell_{\gamma-1}\right|} = \left(\frac{2\gamma - 2\beta - 1}{2\gamma - 1}\right)^{n\ell} = \left(1 - \frac{\beta}{\gamma - 1/2}\right)^{n\ell} \approx \exp\left(-\frac{\beta n \ell}{\gamma}\right).$$

Алгоритм 2 на шаге 5 также проверяет, когда коэффициенты вектора $\mathsf{LSB}(\mathbf{w}, \nu - d)$ не попадают в интервал $\{-(2^{\nu-d} - w \cdot 2^{\nu-\mu+1} - 1), \ldots, 2^{\nu-d} - w \cdot 2^{\nu-\mu+1} - 1\}$. Вероятность этого события, очевидно, зависит от малого вектора ошибки $\mathbf{c}\,(\mathbf{h} + \mathsf{LSB}(\mathbf{As} + \mathbf{h}, \nu - \mu))$, который возникает при упрощении выражения $\mathbf{w} = \mathbf{A} \cdot \mathbf{z} - \mathbf{t} \cdot 2^{\nu-\mu} \cdot \mathbf{c}$ на шаге 4. Вычислим эту вероятность.

Как показано выше, каждый коэффициент вектора ошибки $\mathbf{c}\,(\mathbf{h} + \mathsf{LSB}(\mathbf{As}+\mathbf{h}, \nu-\mu))$ лежит в интервале $\{-(w \cdot 2^{\nu-\mu+1} - 1), \ldots, w \cdot 2^{\nu-\mu+1} - 1\}$. Для каждого такого коэффициента соответствующий коэффициент вектора $\mathsf{LSB}(\mathbf{w}, \nu - d)$ попадает в интервал $\{-(2^{\nu-d} - 1), \ldots, 2^{\nu-d} - 1\}$. Учитывая (эвристически) равномерный характер распределений, в итоге получаем

$$p_2 = \mathsf{P}_{\mathbf{w} \in S^k_{2^{\nu-d}-1}} \left[\|\mathsf{LSB}(\mathbf{w}, \nu - d)\|_\infty < 2^{\nu-d} - w \cdot 2^{\nu-\mu+1} \right] = \left(\frac{2^{\nu-d+1} - w \cdot 2^{\nu-\mu+2} - 1}{2^{\nu-d+1} - 1} \right)^{nk} =$$

$$= \left(1 - \frac{w 2^{\nu-\mu+2}}{2^{\nu-d+1} - 1} \right)^{n \cdot k} \approx \exp\left(-nk \, \frac{w 2^{\nu-\mu+2}}{2^{\nu-d+1} - 1} \right).$$

Таким образом, ожидаемое число повторений функции Sign алгоритма 2 равно

$$\mathbb{E}[\#\text{итераций}] = (p_1 \cdot p_2)^{-1}.$$

3. Атаки и выбор параметров

Безопасность нашей схемы подписи основана на двух классических задачах на решётках — MLWR и MSIS. Будем определять конкретные параметры схемы, основываясь на сложности атак на эти задачи.

Мы работаем с модульными решётками, определёнными над кольцом целых циклотомического расширения, а именно $R = \mathbb{Z}[x]/(x^{256} + 1)$, то есть выбираем $n = 256$. Такое n позволяет осуществлять быструю арифметику в R. Основные параметры, определяющие сложность задач MLWR и MSIS, — это k (задаёт ранг решёток) и ℓ (задаёт размер секретного вектора $\mathbf{s}$).

Решение задачи MLWR сводится к нахождению короткого вектора в q-арной решётке ранга d

$$\Lambda_{\text{MLWR}} = \{\mathbf{x} \in \mathbb{Z}^d : [\text{rot}(\mathbf{A}) \,|\, \mathbb{I} \,|\, \mathbf{t}]\, \mathbf{x} = \mathbf{0} \bmod q\},$$

где $d \leqslant n(\ell + k) + 1$. Мы используем знак $\leqslant$, так как оптимальная атака может не использовать некоторые строки матрицы $[\text{rot}(\mathbf{A}) \,|\, \mathbb{I} \,|\, \mathbf{t}]$. Нужный вектор $\mathbf{x} \in \Lambda_{\text{MLWR}}$ — это $\mathbf{x}_{\text{short}} = [\text{rot}(\mathbf{s})| - \mathbf{t}_{\text{low}}| - 1]$, где $\mathbf{t}_{\text{low}} = \mathbf{As} - \mathbf{t}$ и $\|\mathbf{t}_{\text{low}}\|_\infty \leqslant 2^{\nu-\mu}$. Это «короткий» вектор в решётке Λ_{MLWE}, так как он значительно короче $\sqrt{d} q^{1/nk}$ — границы Минковского для Λ_{MLWR}.

Это классическая «примальная» атака на LWR, сложность которой зависит от времени работы алгоритма BKZ для нахождения вектора длины $\|\mathbf{x}_{\text{short}}\|$. Оценить конкретное время работы BKZ — нетривиальная задача. Для получения значения 104 в таблице — консервативной оценки времени работы BKZ для решения задачи LWR — мы опирались на работу [15] и программный код [16]. Мы не приводим оценку для так называемой «дуальной» атаки на LWR, так как «примальный» метод для наших параметров оказался значительно эффективнее.

Рассмотрим теперь сложность задачи MSIS (так как задача SelfTargetSIS сводится к MSIS и для наших параметров атаки именно на MSIS работают эффективнее, определяющим фактором является сложность MSIS). Наиболее эффективная из всех известных атак на MSIS — нахождение короткого вектора в решётке

$$\Lambda_{\text{MSIS}} = \{\mathbf{x} \in \mathbb{Z}^d : [\text{rot}(\mathbf{A}) \,|\, \mathbb{I}]\, \mathbf{x} = \mathbf{0} \bmod q\}.$$

В отличие от атаки на MLWR, оптимальный алгоритм для задачи MSIS может опустить некоторые *столбцы* матрицы $[\mathrm{rot}(\mathbf{A}) \,|\, \mathbb{I}]$. Решением задачи MSIS считается короткий вектор $\mathbf{x} \in \Lambda_{\mathrm{MSIS}}$ с нормой $\|\mathbf{x}\|_\infty \leqslant \max\{2^{\nu-d+1}, 2(\gamma-\beta)\}$. Для параметров, приведённых в таблице, эти два значения примерно совпадают. Для получения конкретной сложности атаки MSIS мы пользовались стратегией [7, Appendix C]; скрипт, с помощью которого можно получить таблицу, доступен по ссылке `https://crypto-kantiana.com/elena.kirshanova/\#research`.

Предлагаемые параметры цифровой подписи и их уровень безопасности

n	k	ℓ	ν	μ	s	d	γ	$\mathbb{E}$[#итераций]	MSIS (BKZ-b)	MLWR (BKZ-b)
256	3	4	23	19	4	3	1048096	8	93 (320)	104 (357)

В таблице последние два параметра — 93 (соотв. 104) — соответствуют битовой сложности атаки на MSIS с оптимальным размером блока в алгоритме BKZ, равному 320 (соотв. MLWR с оптимальным размером блока 357). В обоих вычислениях полагаем (консервативно), что сложность нахождения короткого вектора в решётке размерности d равна $2^{0{,}292d}$, что асимптотически соответствует сложности алгоритма просеивания.

ЛИТЕРАТУРА

1. *Alkim E., Ducas L., Pöppelmann T., and Schwabe P.* Post-quantum key exchange: A new hope // USENIX Conf. Security Symposium. 2016. P. 327–343.
2. *Adeline L. and Stehlé S.* Worst-case to average-case reductions for module lattices // Des. Codes Cryptography. 2015. V. 75. No. 3. P. 565–599.
3. *Kirshanova E., Kolesnikov N., Malygina E., and Novoselov S.* Проект стандартизации постквантовой цифровой подписи (полная версия). `https://crypto-kantiana.com/main_papers/main_Signature.pdf`.
4. *Fiat A. and Shamir A.* How to prove yourself: Practical solutions to identification and signature problems // CRYPTO'86. LNCS. 1987. V. 263. P. 186–194.
5. *Lyubashevsky V.* Fiat — Shamir with aborts: Applications to lattice and factoring-based signatures // ASIACRYPT'2009. LNCS. 2009. V. 5912. P. 598–616.
6. *Bai S. and Galbraith S. D.* An improved compression technique for signatures based on learning with errors // Topics in Cryptology — CT-RSA 2014. LNCS. 2014. V. 8366. P. 28–47.
7. *Ducas L., Kiltz E., Lepoint T., et al.* CRYSTALS-Dilithium: A lattice-based digital signature scheme // IACR Trans. Cryptographic Hardware and Embedded Systems. 2018. No. 1. P. 238–268.
8. *Alkim E., Bindel N., Buchmann J., et al.* Revisiting TESLA in the quantum random oracle model // PQCrypto 2017. LNCS. 2017. V. 10346. P. 143–162.
9. *D'Anvers J.-P., Karmakar A., Roy S. S., and Vercauteren F.* Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM // Progress in Cryptology — AFRICACRYPT 2018. Springer, 2018. P. 282–305.
10. *Banerjee A., Peikert C., and Rosen A.* Pseudorandom functions and lattices // Ann. Intern. Conf. Theory and Appl. of Cryptographic Techniques. Springer, 2012. P. 719–737.
11. *Regev O.* On lattices, learning with errors, random linear codes, and cryptography // J. ACM. 2005. V. 56. No. 6. P. 84–93.
12. *Bogdanov A., Guo S., Masny D., et al.* On the hardness of learning with rounding over small modulus // Theory of Cryptography. LNCS. 2016. V. 9562. P. 209–224.
13. *Ajtai M.* Generating hard instances of lattice problems (extended abstract) // Proc. 28th Ann. ACM Symp. Theory Computing. 1996. P. 99–108.

14. *Kiltz E., Lyubashevsky V., and Schaffner C.,* A concrete treatment of Fiat — Shamir signatures in the quantum random-oracle model // Adv. Cryptology — EUROCRYPT 2018. Springer, 2018. P. 552–586.
15. *Albrecht M. R., Göpfert F., Virdia F., and Wunderer T.* Revisiting the expected cost of solving uSVP and applications to LWE // ASIACRYPT 2017. LNCS. 2017. V. 10624. P. 297–322.
16. *Albrecht M. R., Curtis B. R., Deo A., et al.* Estimate all the {LWE, NTRU} schemes! // SCN 2018. LNCS. 2018. V. 11035. P. 351–367.

УДК 512.64, 519.21, 519.72 DOI 10.17223/2226308X/13/15

КОНСТРУКЦИИ НЕЭНДОМОРФНЫХ СОВЕРШЕННЫХ ШИФРОВ

Н. В. Медведева, С. С. Титов

Исследуются совершенные по Шеннону (абсолютно стойкие к атаке по шифртексту) шифры. Получены достаточные условия того, что таблицы зашифрования неэндоморфных (эндоморфных) совершенных шифров не содержат латинских прямоугольников (квадратов). Приведён пример таких конструкций.

Ключевые слова: *совершенные шифры, эндоморфные шифры, неэндоморфные шифры.*

Рассмотрим вероятностную модель Σ_B шифра [1]. Пусть X, Y — конечные множества соответственно шифрвеличин и шифробозначений, с которыми оперирует некоторый шифр замены, K — множество ключей, причём $|X| = \lambda$, $|Y| = \mu$, $|K| = \pi$, где $\lambda > 1$, $\mu \geqslant \lambda$. Это означает, что открытые и шифрованные тексты представляются словами (ℓ-граммами, $\ell \geqslant 1$) в алфавитах X и Y соответственно. Согласно [2, 3], под *шифром* Σ_B будем понимать совокупность множеств правил зашифрования и правил расшифрования с заданными распределениями вероятностей на множествах открытых текстов и ключей. Шифры, для которых апостериорные вероятности открытых текстов совпадают с их априорными вероятностями, называются *совершенными*.

Описание эндоморфных ($\lambda = \mu$) с минимально возможным числом ключей ($|K| = |Y|$) совершенных шифров даёт теорема Шеннона, таблица зашифрования таких шифров — это латинский квадрат из равновероятных подстановок зашифрования [1].

Для неэндоморфных ($\lambda < \mu$) минимальных совершенных шифров характерно большое многообразие таблиц зашифрования: они не сводятся только к латинским прямоугольникам размера $\mu \times \lambda$ [4]. Для $\lambda = 2$, например, таблицы зашифрования могут быть составлены и из неравновероятных инъекций. Однако если все ключи равновероятны, то данный совершенный шифр является выпуклой оболочкой латинских прямоугольников, содержащихся в его таблице зашифрования, согласно аналогу теоремы Биркгофа [5]. Если $\lambda > 2$, то, даже для равновероятных инъекций зашифрования, неэндоморфный совершенный шифр может не содержать в своей таблице зашифрования латинских прямоугольников $\mu \times \lambda$ [6].

Таким образом, при $\mu > \lambda$ возникает естественная задача описания минимальных по включению (т. е. шифров, содержащих минимально возможное множество ключей зашифрования с ненулевыми вероятностями) совершенных шифров, не сводящихся к латинским прямоугольникам размера $\mu \times \lambda$, которые можно рассматривать как непосредственное обобщение теоремы Шеннона. Данную задачу можно трактовать как задачу описания выпуклого полиэдра, соответствующего совершенным шифрам, через нахождение его вершин [5].

Подходом к решению такой задачи могут быть конструкции таблиц зашифрования, не содержащих латинских прямоугольников. Первым этапом решения задачи описания минимальных (по включению) совершенных шифров является построение таких конструкций для равновероятных ключей. При этом полезны достаточные условия отсутствия в таких таблицах латинских прямоугольников, тем более что в некоторых случаях эти условия оказываются необходимыми и достаточными.

Более того, эти условия могут быть частью общей конструкции искомых таблиц зашифрования, так как любые два её столбца можно рассматривать как шифр с $\lambda = 2$, к которому применим аналог теоремы Биркгофа [5, 7]. Тем самым построение таблицы зашифрования при $\lambda > 2$ можно трактовать как расширение таблицы с $\lambda = 2$. Для равновероятных ключей латинские прямоугольники $\mu \times 2$ всегда содержатся в таблице зашифрования с $\lambda \geqslant 3$ и поэтому достаточные условия должны включать в себя не менее трёх столбцов таблицы.

Пример 1. Пусть таблица зашифрования с $\lambda = 3 = |X|$, где $X = \{x_1, x_2, x_3\}$ — множество (алфавит) шифрвеличин, содержит строки с ключами k_i, k_j, k_m с вероятностями p_i, p_j, p_m соответственно и с шифробозначениями $a, b, c, u, v, w \in Y$, где Y — множество шифробозначений, $|Y| = \mu$:

k	x_1	x_2	x_3	P
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
k_i	a	b	u	p_i
k_j	v	b	c	p_j
k_m	a	w	c	p_m
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$

Здесь шифробозначения a, b и c не входят в другие строки этой таблицы. Тогда данная таблица не может содержать латинских прямоугольников размеров $\mu \times 3$.

Действительно, если такой прямоугольник имеется, то он содержит либо строку ключа k_i, либо строку ключа k_m, так как в столбце для шифрвеличины x_1 шифробозначение a больше не встречается. Если он содержит строку k_i, то строка ключа k_m в него не входит, поэтому он должен содержать строку k_j, так как в столбце x_3 шифробозначение c больше не встречается. Тогда в столбце x_2 шифробозначение b будет встречаться дважды, что невозможно в латинском прямоугольнике.

Если же латинский прямоугольник содержит строку ключа ключа k_m, то строка ключа k_i в него не входит из-за шифробозначения a в столбце x_1. Тогда в нём содержится строка k_j из-за шифробозначения b в столбце x_2. Следовательно, в столбце x_3 шифробозначение c будет встречаться дважды, что также невозможно в латинском прямоугольнике.

Пример 1 иллюстрирует утверждение 1.

Утверждение 1. Пусть a, b, c — различные шифробозначения, $X = \{x_1, x_2, x_3\}$ — множество шифрвеличин. При этом:

1) множество K ключей разбито на два непересекающихся подмножества K_1 и K_2, т. е. $K = K_1 \cup K_2$ и $K_1 \cap K_2 = \varnothing$;
2) для любого ключа $k \in K_1$ шифрвеличина x_2 на ключе k зашифровывается в шифробозначение b, т. е. $e_k(x_2) = b$;
3) существует такой ключ $k \in K_1$, что шифрвеличина x_1 на ключе k зашифровывается в шифробозначение a, т. е. $e_k(x_1) = a$;
4) для любого ключа $k \in K_2$ шифрвеличина x_2 на ключе k зашифровывается в шифробозначение, отличное от шифробозначения b, т. е. $e_k(x_2) \neq b$;

5) существует единственный ключ k из K_2, на котором шифрвеличина x_1 зашифровывается шифробозначением a, а шифрвеличина x_3 — шифробозначением c, т. е. $e_k(x_1) = a$ и $e_k(x_3) = c$.

Тогда таблица зашифрования не содержит латинских прямоугольников $\mu \times 3$.

Определение 1. Ключи k' и k'' эквивалентны по шифрвеличине x_i, если x_i на ключах k' и k'' зашифровывается в одно и то же шифробозначение, т. е.

$$k' \underset{i}{\equiv} k'' \;\Leftrightarrow\; e_{k'}(x_i) = e_{k''}(x_i).$$

Определение 2. Попарно различные ключи $k_1, k_2, k_3, \ldots, k_{n-1}, k_n$ образуют цикл длины n, если выполняются условия

$$k_1 \underset{i_2}{\equiv} k_2 \underset{i_3}{\equiv} k_3 \underset{i_4}{\equiv} \ldots \underset{i_{n-1}}{\equiv} k_{n-1} \underset{i_n}{\equiv} k_n \underset{i_1}{\equiv} k_1,$$

где $i_2 \neq i_3$, $i_3 \neq i_4$, $\ldots$, $i_{n-1} \neq i_n$, $i_n \neq i_1$.

Обозначим через $[k]_i$ смежный класс ключа k по отношению эквивалентности $\underset{i}{\equiv}$:

$$[k]_i = \{k' \in K : e_{k'}(x_i) = e_k(x_i)\}.$$

Утверждение 2. Пусть в таблице зашифрования с $\lambda = 3 = |X|$ ключи k_1, k_2, k_3 образуют цикл длины три:

$$k_1 \underset{i_2}{\equiv} k_2 \underset{i_3}{\equiv} k_3 \underset{i_1}{\equiv} k_1,$$

где i_1, i_2, i_3 — попарно различны, и при этом $[k_3]_{i_3} \setminus [k_1]_{i_2} = \{k_3\}$. Тогда инъекция ключа k_1 не может быть строкой никакого латинского прямоугольника. Кроме того, если

$$[k_3]_{i_1} \subset \left([k_1]_{i_2} \cup [k_2]_{i_2}\right),$$

то таблица зашифрования не содержит латинских прямоугольников.

Из утверждения 2 следует достаточное условие отсутствия латинских квадратов в таблице зашифрования эндоморфного совершенного шифра с $\lambda \geqslant 3$.

Утверждение 3. Если в таблице зашифрования ключи $k_1, k_2, \ldots, k_n$ образуют цикл нечётной длины, то данная таблица не содержит латинских прямоугольников.

Таким образом, на основе отношения эквивалентности на множестве ключей получены достаточные условия того, что в таблице зашифрования неэндоморфных совершенных шифров отсутствуют латинские прямоугольники. В частности, получены достаточные условия того, что таблицы зашифрования эндоморфных совершенных шифров не содержат латинских квадратов.

ЛИТЕРАТУРА

1. *Шеннон К.* Теория связи в секретных системах // Работы по теории информации и кибернетике. М.: Наука, 1963. С. 333–402.
2. *Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В.* Основы криптографии. М.: Гелиос АРВ, 2001.
3. *Зубов А. Ю.* Совершенные шифры. М.: Гелиос АРВ, 2003.
4. *Медведева Н. В., Титов С. С.* Аналоги теоремы Шеннона для эндоморфных неминимальных шифров // Прикладная дискретная математика. Приложение. 2016. № 9. С. 62–65.

5. *Медведева Н. В., Титов С. С.* Описание неэндоморфных максимальных совершенных шифров с двумя шифрвеличинами // Прикладная дискретная математика. 2015. № 4 (30). С. 43–55.
6. *Медведева Н. В., Титов С. С.* Геометрическая модель совершенных шифров с тремя шифрвеличинами // Прикладная дискретная математика. Приложение. 2019. № 12. С. 113–116.
7. *Birkhoff G. D.* Tres observations sobre el algebra lineal // Revista Universidad Nacional Tucuman. 1946. Ser. A. V. 5. P. 147–151.

УДК 519.7 DOI 10.17223/2226308X/13/16

ПОСТРОЕНИЕ РАЗЛИЧИТЕЛЕЙ ДЛЯ ИТЕРАТИВНЫХ БЛОЧНЫХ ШИФРОВ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ

А. А. Перов, А. И. Пестунов

Предлагается новый универсальный подход к построению атак-различителей на итеративные блочные шифры, подразумевающий использование нейронных сетей, предназначенных для классификации растровых изображений. Описываются два метода, основанных на идее представления шифртекстов после разного числа раундов шифрования в виде растровых изображений с последующим обучением нейронной сети распознавать эти изображения. Показано, что для ряда современных блочных шифров предлагаемый подход более эффективен, чем универсальные различители, основанные на статистических тестах.

Ключевые слова: *блочный шифр, машинное обучение, нейронная сеть, статистический анализ, атака-различитель.*

В работе предлагается новый универсальный подход к построению атак-различителей на итеративные блочные шифры, где используются нейронные сети, предназначенные для классификации растровых изображений. Идея данного подхода возникла в результате наблюдения того, что преобразованный в растровое изображение (графический эквивалент) шифртекст имеет различную текстуру (паттерн) в зависимости от числа раундов. При этом с ростом числа раундов такая текстура становится менее выраженной и приближается к случайной.

В рамках этого подхода предлагаются два метода: «эталонный» и метод соседних раундов. В первом нейронная сеть используется для выявления различий в текстурах графических эквивалентов шифртекста при различном числе раундов и эталонной последовательности, неотличимой от случайных чисел. Второй метод предполагает выявление различий в текстурах графических эквивалентов соседних раундов и, что является его достоинством, не требует наличия эталонной последовательности, однако, забегая вперед, отметим, что «эталонный» метод оказался немного более эффективен. В экспериментах в качестве эталонной последовательности использован шифртекст полнораундового шифра AES256.

Для реализации предлагаемых методов необходимо выполнить процесс обучения свёрточной нейронной сети на графических эквивалентах шифртекстов (алгоритм 1).

Для формирования выборки выполняется шифрование на разном числе раундов, что даёт выборку выходных последовательностей блочных шифров с разными статистическими свойствами. На шагах 2–3 алгоритма 1 с помощью криптографической программной библиотеки «УНИБЛОКС-2015» выполняется шифрование в режиме

Алгоритм 1. Обучение нейронной сети для распознавания шифртекста после заданного числа раундов

1: **Функция** ОбучитьНейроннуюСеть($Cipher$, r, M)
   // *$Cipher$ — итеративный блочный шифр;*
   // *r — число раундов шифра;*
   // *M — размер обучающего множества.*
2: Сгенерировать M выборок с помощью шифра AES256 и получить $\tilde{\mathcal{Y}}^{\text{rand}} = (\tilde{y}_1^{\text{rand}}, \ldots, \tilde{y}_M^{\text{rand}})$.
3: Сгенерировать M выборок с помощью шифра $Cipher$ и получить $\tilde{\mathcal{Y}}^r = (\tilde{y}_1^r, \ldots, \tilde{y}_M^r)$.
4: Преобразовать множества выборок $\tilde{\mathcal{Y}}^{\text{rand}}$ и $\tilde{\mathcal{Y}}^r$ в изображения $\mathcal{Y}^{\text{rand}} = (y_1^{\text{rand}}, \ldots, y_M^{\text{rand}})$ и $\mathcal{Y}^r = (y_1^r, \ldots, y_M^r)$.
5: Обучить нейронную сеть различать изображения из обучающих выборок $\mathcal{Y}^{\text{rand}}$ и $\mathcal{Y}^r$.
6: **Вернуть** НейроннаяСеть^r(image), которая относит image к 0 (случайному) или 1 (r-раундовому шифртексту).

счётчика (CTR) [1]. В качестве входной последовательности для блочного шифра использованы последовательные числа 0, 1, 2, 3 и т. д.

На шаге 4 шифртексты преобразуются в формат растровых графических изображений с помощью разработанной программной утилиты на языке C++.

В процессе обучения нейронная сеть запоминает основные паттерны, характерные для шифртекстов на разном числе раундов, которые использует для последующей категоризации.

После обучения свёрточной нейронной сети выполняется распознавание шифртекстов. В зависимости от выбранного метода на контрольной выборке нейронная сеть сравнивает шифртексты на соседних раундах шифрования или шифртексты выбранного раунда с «эталоном» — полнораундовым AES256, подсчитывает процент верных решений при определении принадлежности элемента контрольной выборки к тому или иному множеству. С увеличением числа раундов шифрования и соответственно улучшением статистических свойств модель увеличивает значение E — число ошибок, допущенных моделью при определении принадлежности к тому или иному раунду на контрольной выборке. Алгоритм 2 описывает атаку-различитель.

С увеличением числа раундов ошибка при различении шифртекстов и эталонной случайной последовательности возрастает и стремится к 0,5. Алгоритм 3 описывает схему проведения экспериментов, в которых анализируется способность предлагаемого подхода различать шифртексты на примере «эталонного» метода.

Обоснование эффективности метода соседних раундов выполняется аналогично, за исключением того, что генерируются не выборки шифра AES256, а выборки x^r и x^{r+1}. Размер обучающей выборки выбирается нейронной сетью в зависимости от исходных параметров (в большей степени — от размера партии, то есть количества изображений, в которых нейронная сеть выполняет поиск общих признаков). Экспериментально определено, что 500 шифртекстов достаточно для обучения. Контрольная выборка составляет 20 % от обучающей. При этом процент ошибок нейронной сети на первых раундах выше (так как соседние раунды различимы между собой много меньше, чем при сравнении с эталонным шифртекстом), однако при достижении числа раундов, при котором обеспечиваются удовлетворительные статистические свойства, ошибка также сводится к 0,5.

Алгоритм 2. Атака-различитель

Функция РАСПОЗНАТЬШИФРТЕКСТ(x, $Cipher$, r)
// x — запрошенная в шифровальном устройстве выборка (генерируется в режиме CTR в сценарии chosen-plaintext attack);
// $Cipher$ — итеративный блочный шифр;
// r — число раундов шифра.
Выбрать размер обучающей выборки M.
НейроннаяСеть^r(image): = ОБУЧИТЬНЕЙРОННУЮСЕТЬ($Cipher$, r, M).
Представить выборку x в виде изображения image.
Result := НейроннаяСеть^r(image).
Если Result = 0, **то**
  вернуть «Выборка случайная»,
иначе
  вернуть «Выборка сгенерирована r-раундовым шифром».

Алгоритм 3. Схема проведения экспериментов

Функция ВЫЧИСЛИТЬОШИБКУ($Cipher$, r)
Выбрать размер обучающей выборки M.
НейроннаяСеть^r(image) := ОБУЧИТЬНЕЙРОННУЮСЕТЬ($Cipher$, r, M).
Выбрать количество контрольных выборок N.
Сгенерировать N контрольных выборок с помощью шифра AES256 и получить $\tilde{\mathcal{X}}^{\text{rand}} = (\tilde{x}_1^{\text{rand}}, \ldots, \tilde{x}_N^{\text{rand}})$.
Сгенерировать N контрольных выборок с помощью шифра $Cipher$ и получить $\tilde{\mathcal{X}}^r = (\tilde{x}_1^r, \ldots, \tilde{x}_N^r)$.
Преобразовать множества $\tilde{\mathcal{X}}^{\text{rand}}$ и $\tilde{\mathcal{X}}^r$ в изображения $\mathcal{X}^{\text{rand}} = (x_1^{\text{rand}}, \ldots, x_N^{\text{rand}})$ и $\mathcal{X}^r = (x_1^r, \ldots, x_N^r)$.
Экспериментально определить ошибки первого и второго рода:
$E_0 = \#\{x_i^{\text{rand}} : \text{НейроннаяСеть}^r(x_i^{\text{rand}}) = 1\}$, $E_1 = \#\{x_i^r : \text{НейроннаяСеть}^r(x_i^r) = 0\}$.
Вернуть E_0, E_1.

ЛИТЕРАТУРА

1. *Пестунов А. И., Перов А. А.* Программная библиотека для статистического анализа итеративных блочных шифров // Информационное противодействие угрозам терроризма. 2015. № 24. С. 197–202.

УДК 003.26 DOI 10.17223/2226308X/13/17

О СКРЫТОМ КОМПАКТНОМ СПОСОБЕ ХРАНЕНИЯ ДАННЫХ[1]

В. А. Романьков

Предлагается принципиально новый способ компактного хранения данных в скрытом виде. Каждое из этих данных может быть извлечено единообразным способом. Приводится сравнение с другими возможными способами такого хранения.

Ключевые слова: *данные, хранение, скрытость, компактность, доступ.*

[1]Исследование поддержано Программой фундаментальных научных исследований СО РАН I. 1.1.4, проект № 0314-2019-0004.

Допустим, имеются данные $a_i,\ i = 1, \ldots, n$, каждое из которых представляет собой бинарную строку длины m, то есть $a_i \in \{0,1\}^m$. Предположим, что эти данные необходимо хранить в компактном скрытом виде X, имея простой единообразный способ извлечения из X любого из них. Такая проблема может возникнуть при хранении в облаке, а также в других случаях защищённого хранения. Например, её решение может иметь значение для поисковых систем. Может также возникнуть потребность хранить данные, разбитые на подмножества по типу данных или их принадлежности.

В работе представлен принципиально новый способ решения указанной проблемы. Но сначала рассмотрим некоторые напрашивающиеся идеи такого хранения.

Хранение в зашифрованном виде. Можно использовать какую-нибудь систему шифрования и хранить данные в виде $E_k(a_i),\ i = 1, \ldots, m$, где E_k — функция зашифрования с ключом k. Такой способ, конечно, обеспечивает свойство «скрытости», но непонятно, как обеспечить компактность. Нужно хранить ключи, зашифрованные данные могут быть существенно большего объёма и т. п. Эффективность такого способа хранения зависит от свойств выбранной системы шифрования. Если использовать один ключ для всех данных, то ключ расшифрования не может быть делегирован пользователю, которому необходимо извлечь только какие-то данные, к которым ему может быть предоставлен доступ. Возможно, что указанные недостатки могут быть устранены, но для этого требуется отдельное исследование.

Использование Китайской теоремы об остатках. Напомним эту известную теорему (см., например, [1]). Предположим, что целые числа $m_1, \ldots, m_n \geqslant 2$ попарно взаимно просты. Тогда система сравнений $x = b_i \pmod{m_i},\ i = 1, \ldots, n$, всегда имеет решение X, которое находится следующим образом. Полагаем $M_i = M/m_i$, где $M = \prod_{i=1}^{n} m_i$. В силу попарной взаимной простоты модулей m_i существуют числа L_i, для которых $L_i M_i = 1 \pmod{m_i}$. Тогда число $X = \sum_{i=1}^{n} b_i L_i M_i$ является решением системы. Более того, решением является любое целое число Y, такое, что $Y = X \pmod{M}$, и других решений нет.

Покажем, как можно организовать скрытое компактное хранение данных, используя эту теорему. Сопоставим каждой последовательности $a_i \in \{0,1\}^m$ натуральное число b_i, получающееся, если считать запись a_i выражением числового значения в двоичной системе. Выберем попарно взаимно простые модули $m_i > b_i,\ i = 1, \ldots, n$. Храним данные b_i в виде решения X рассмотренной выше системы. Для извлечения b_i вычисляем X по модулю m_i. Скрытость и возможность единообразного извлечения данных обеспечены, чего нельзя сказать в общем случае о компактности этого способа хранения данных. Действительно, если числа b_i достаточно большие, то и модули m_i большие. Если данных достаточно много, то число M (а также числа M_i) становится нереально большим. Это не только замедлит вычисления, но может сделать само хранение практически нереализуемым.

Хранение в «магическом» квадрате. Пусть $d = d_1, d_2, \ldots, d_i, \ldots$ — последовательность целых чисел. Дискретная производная d' этой последовательности определяется формулой

$$d' = d_2 - d_1, d_3 - d_2, \ldots, d_{i+1} - d_i, \ldots$$

Очевидно, что производная аддитивна. Для любой последовательности определяется дискретный интеграл $f = f_1, f_2, \ldots, f_i, \ldots,\ f' = d$. Интеграл определяется с точностью

до произвольной константы c формулой

$$f_i = c + \sum_{j=1}^{i-1} d_j.$$

При выборе $f_1 = c$ все остальные компоненты интеграла f определяются однозначно. Обозначим $f = I(d, c)$.

Пусть дана постоянная последовательность $c^{(0)} = c, \ldots, c, \ldots$ Обозначим $c^{(1)} = I(c^{(0)}, c)$, $c^{(2)} = I(c^{(1)}, c)$, $\ldots$ Тогда значение k-й производной от $c^{(k)}$ равно $c^{(0)}$. Записываем это в виде $[c^{(k)}, z; k] = c$. При $i < k$ имеем $[c^{(i)}, x; k] = 0$. Мы несколько упростили запись, поставив в правую часть значение постоянной последовательности, а не саму эту последовательность. Символ z введён для различия в дальнейшем переменных интегрирования.

Пусть имеется n ненулевых целых чисел $c_1, \ldots, c_n$. Для каждого i вычислим интеграл $c_i^{(n-i)}$. Построим квадратную таблицу со стороной $n + 1$. Запишем в нижней строке первые $n + 1$ элементов интеграла $c_i^{(n-i)}$. Получим набор $(c_{i,1,1}, \ldots, c_{i,1,n+1})$.

Затем для каждого элемента нижней строки $c_{i,1,j}$ определим постоянную последовательность $c_{i,1,j}^{(0)}$ и вычислим для неё интеграл $c_{i,1,j}^{(i)}$. Тогда $[c_{i,1,j}^{(i)}, y; i] = c_{i,1,j}$. Поставим в соответствующий столбец таблицы первые $n + 1$ элементов этого интеграла.

Пусть X означает сумму всех таких таблиц.

Теорема 1. Справедлива формула

$$[X, y; i, z; n - i]_{1,1} = c_i.$$

Формула означает, что i-кратное дифференцирование столбцов с последующим $(n-i)$-кратным дифференцированием нижней строки даёт в левом нижнем углу значение c_i.

Замечание 1.

— Если записывать все последовательности, а не только их начальные отрезки, то в результате проведённых операций получится бесконечная вправо и вверх таблица, в которой во всех клетках будет стоять соответствующая константа.

— Так как натуральное отображение вида $\mathbb{Z} \to \mathbb{Z}_r$ является гомоморфизмом, то можно выбрать $r > \max(b_i : i = 1, \ldots, n)$ и вести все построения над элементами кольца $\mathbb{Z}_r$.

Мы не приводим доказательство теоремы; дадим только небольшой иллюстрирующий пример.

Пример 1. Пусть $b_1 = 12$, $b_2 = 7$, $b_3 = 3$. Вычисления ведём в кольце $\mathbb{Z}_{13}$. Запишем в общем виде начальные интервалы длины 4 от константы a:

$$a_4^{(1)} = (a, 2a, 3a, 4a),\ a_4^{(2)} = (a, 2a, 4a, 7a),\ a_4^{(3)} = (a, 2a, 4a, 8a).$$

Запишем квадраты для $b_1 = 12$, $b_2 = 7$, $b_3 = 3$:

$$\left|\begin{array}{cccc} 9 & 5 & 10 & 7 \\ 10 & 7 & 1 & 2 \\ 11 & 9 & 5 & 10 \\ 12 & 11 & 9 & 5 \end{array}\right|, \quad \left|\begin{array}{cccc} 10 & 7 & 1 & 5 \\ 2 & 4 & 8 & 1 \\ 1 & 2 & 4 & 7 \\ 7 & 1 & 2 & 10 \end{array}\right|, \quad \left|\begin{array}{cccc} 11 & 9 & 7 & 5 \\ 12 & 11 & 10 & 9 \\ 6 & 12 & 5 & 11 \\ 3 & 6 & 9 & 12 \end{array}\right|.$$

Тогда

$$X = \begin{vmatrix} 4 & 8 & 5 & 4 \\ 11 & 9 & 6 & 12 \\ 5 & 10 & 1 & 2 \\ 9 & 5 & 7 & 1 \end{vmatrix}.$$

Для проверки проводим вычисления, записывая только те строки и столбцы, значения в которых полностью определяются заданными квадратами:

$$[X, y; 1] = \begin{vmatrix} 6 & 12 & 12 & 5 \\ 6 & 12 & 5 & 10 \\ 9 & 5 & 7 & 1 \end{vmatrix}, \quad [X, y; 2] = \begin{vmatrix} 0 & 0 & 7 & 8 \\ 10 & 7 & 11 & 9 \end{vmatrix}, \quad [X, y; 3] = \begin{vmatrix} 3 & 6 & 9 & 12 \end{vmatrix},$$

$$[X, y; 1, z; 3]_{1,1} = 12, \quad [X, y; 2, z; 2]_{1,1} = 7, \quad [X, y; 3, z; 1]_{1,1} = 3.$$

ЛИТЕРАТУРА

1. *Романьков В. А.* Введение в криптографию. М.: Форум, 2012.

УДК 519.17 DOI 10.17223/2226308X/13/18

ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ ОДНОГО КЛАССА КРИПТОАЛГОРИТМОВ НА ОСНОВЕ ОБОБЩЕНИЯ СЕТЕЙ ФЕЙСТЕЛЯ

В. М. Фомичёв, Д. А. Бобровский, А. М. Коренева

Представлены результаты экспериментальных исследований производительности алгоритма 256-3 (с блоком 256 бит и тремя функциями обратной связи), предложенного российскими исследователями в 2018 г. Производительность 256-3 оценивается величиной 24,57 циклов на байт. Проведено сравнение с известными блочными шифрами, получены оценки для программных реализаций алгоритмов на языке программирования C++ с использованием библиотеки Crypto++. Установлено, что производительность 256-3 от 1,2 до 2,6 раз превышает производительность алгоритмов «Магма» (ГОСТ 28147-89), «Кузнечик» (ГОСТ 34.12-2018), SEED, HIGHT, Camellia-256, Kalyna-256/256, MARS-256, CAST-256, что указывает на положительные (с позиции синтеза) эксплуатационные качества алгоритма 256-3.

Ключевые слова: *блочные шифры, производительность шифрования, 256-3, ГОСТ 28147-89, ГОСТ 34.12-2018, «Магма», «Кузнечик», AES, Rijndael, SEED, SM4, HIGHT, Camellia, Kalyna, MARS, CAST, RC6, Crypto++.*

Введение

Развитие информационных технологий и необходимость защиты информации определяют актуальность разработки новых криптографических алгоритмов, соответствующих современным требованиям к криптографической стойкости и эксплуатационным качествам. Для обеспечения конфиденциальности информации при её передаче, обработке и хранении требуются алгоритмы с высокой производительностью и варьируемыми параметрами (размерами длины ключа и блока) в зависимости от типа задачи.

С целью увеличения производительности блочного шифрования в [1, 2] исследован класс регистровых преобразований $R(n, r, m)$, реализуемых автономными регистрами сдвига длины n над множеством $V_r = \{0, 1\}^r$ с m обратными связями, $n > m \geqslant 1$. Идея

увеличения производительности состоит в увеличении размера блока данных при относительно небольшом увеличении числа обратных связей. Предложены способы построения биективных раундовых функций с блоками от 256 до 1056 бит, при которых координатные функции шифрующих подстановок нелинейные и реализуют полное перемешивание битов входного блока. На примере алгоритма 256-3 (с блоком 256 бит и тремя функциями обратной связи, аналогичными функции усложнения ГОСТ 28147-89) показано, что построенные алгоритмы превышают по производительности алгоритмы на основе классической сети Фейстеля. Экспериментально установлено, что производительность 32-раундового алгоритма 256-3 в два раза превышает производительность ГОСТ 28147-89.

В работе представлены новые результаты экспериментальных исследований производительности алгоритма 256-3, проведено сравнение с известными блочными шифрами, которые являются международными, отраслевыми и национальными стандартами, а также рекомендованными международной организацией по стандартизации (ISO) [3].

1. Схема раундовой функции алгоритма 256-3

Опишем принцип построения раундовой функции g алгоритма 256-3 [1, 2] (схема на рис. 1). Для фиксированного раунда обозначим:

$X = (X_0, \ldots, X_7)$ — входной блок раунда, $X \in V_{256}, X_k \in V_{32}, 0 \leqslant k \leqslant 7$;

$Y = (Y_0, \ldots, Y_7)$ — выходной блок раунда, $Y \in V_{256}, Y_k \in V_{32}, 0 \leqslant k \leqslant 7$;

S — сумма по модулю 2^{32} нескольких подблоков входного блока, представленных числами из кольца вычетов $\mathbb{Z}_{2^{32}}$;

q_j — раундовый ключ, использующийся при вычислении значения функции обратной связи с номером $j \in \{1, 2, 3\}$;

$\oplus$ — сложение по модулю 2, $\boxplus$ — сложение по модулю 2^{32}.

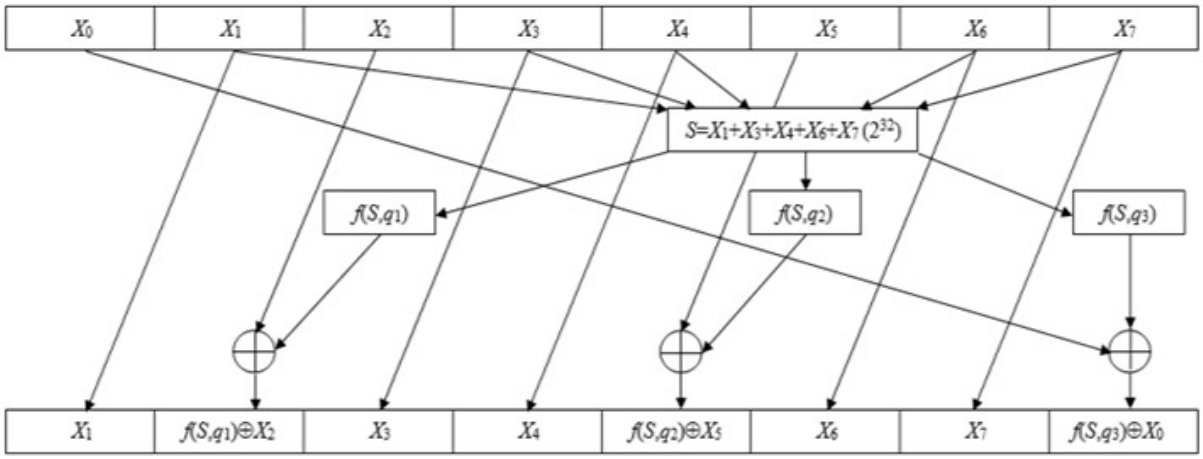

Рис. 1. Схема раундовой подстановки g алгоритма 256-3

Раундовая подстановка g алгоритма 256-3 и обратная к ней определены формулами

$$g(q_1, q_2, q_3)(X_0, \ldots, X_7) = (X_1, f(S, q_1) \oplus X_2, X_3, X_4, f(S, q_2) \oplus X_5, X_6, X_7, f(S, q_3) \oplus X_0),$$
$$g^{-1}(q_1, q_2, q_3)(Y_0, \ldots, Y_7) = (f(S', q_3) \oplus Y_7, Y_0, f(S', q_1) \oplus Y_1, Y_2, Y_3, f(S', q_2) \oplus Y_4, Y_5, Y_6),$$

где $S = X_1 \boxplus X_3 \boxplus X_4 \boxplus X_6 \boxplus X_7$ и $S' = Y_0 \boxplus Y_2 \boxplus Y_3 \boxplus Y_5 \boxplus Y_6$.

Регистр сдвига g имеет три идентичных обратных связи $X_2 \oplus f(S, q_1)$, $X_5 \oplus f(S, q_2)$, $X_0 \oplus f(S, q_3)$, каждая из которых построена по принципу раундовой функции блочного

шифра «Магма» (ГОСТ 34.12-2018). Функция f имеет вид $f(S, q_j) = T^{11}(W_{8,4}(S \boxplus q_j))$, где $\boxplus q_j$ — сложение с раундовым ключом q_j по модулю 2^{32}; $W_{8,4}$ — преобразование V_{32}, реализуемое восемью 4-битовыми s-боксами алгоритма «Магма»; T^{11} — преобразование циклического левого сдвига на 11 бит.

2. Описание эксперимента и результаты исследования

Проведено сравнение производительности алгоритма 256-3 с производительностью известных блочных шифров: AES-256, «Магма», «Кузнечик», SEED, SM4, HIGHT, Camellia-256, Kalyna-256/256, MARS-256, CAST-256, RC6-256. Программная реализация алгоритмов шифрования 256-3 и «Кузнечик» выполнена на языке программирования C++ с использованием кроссплатформенной бесплатной библиотеки Crypto++ 8.2 с открытым исходным кодом [4]. Для остальных алгоритмов использованы реализации из библиотеки Crypto++. Выбор данной библиотеки обусловлен большим количеством криптографических алгоритмов и высокой скоростью реализаций в сравнении с другими криптографическими библиотеками (на языках Python, C, C ++).

Эксперименты проведены на ПЭВМ с процессором Intel(R) Core(TM) i5-7600 с постоянной тактовой частотой $U = 3{,}89\,\text{ГГц}$, архитектура операционной системы 64-битная (x64). Расширение системы команд AES-NI отключено. Оптимизация программного кода — /O2.

Для каждого из алгоритмов выполнено зашифрование открытого текста длиной $L = 268435456$ байт в режиме простой замены, ключ вырабатывался программным датчиком случайных чисел в составе библиотеки Crypto++. Время t, затраченное на зашифрование, измерялось с помощью системных часов реального времени стандартной библиотеки chrono. Затем рассчитывалось количество мебибайт (2^{30} байт) обработанного открытого текста в секунду (МиБ/с) и независимая от частоты процессора характеристика производительности шифра — количество циклов на байт (CpB), согласно формуле $\text{CpB} = tU/L$. Результаты экспериментов приведены в таблице в порядке убывания производительности.

Производительность шифров

Шифр	Число раундов	CpB	МиБ/с
AES-256	14	7,92497	468,114
SM4	32	14,6612	253,035
RC6-256	20	15,2273	243,628
256-3	32	24,5674	151,005
MARS	32	29,2092	127,008
Camellia-256	24	31,247	118,725
CAST-256	48	31,4735	117,871
«Магма»	32	48,2291	76,9202
Kalyna-256/256	14	50,7198	73,1429
SEED	16	55,022	67,4239
«Кузнечик»	10	62,2676	59,5782
HIGHT	32	64,3055	57,6901

Выводы

Результаты показали, что производительность реализации 256-3 ниже производительности AES-256, SM4, RC6-256 в 3, 1,68 и 1,61 раз соответственно. Это связано с высокой скоростью выполнения примитивных операций, заложенных в данные алгоритмы, на системах с архитектурой Intel IA-64. В то же время производительность

256-3 в 1,2–2,6 раз превышает производительность алгоритмов «Магма» (ГОСТ 34.12-2018), «Кузнечик» (ГОСТ 34.12-2018), SEED, HIGHT, Camellia-256, Kalyna-256/256, MARS-256, CAST-256, что указывает на положительные (с позиции синтеза) эксплуатационные качества алгоритма 256-3 и представляет данный алгоритм перспективным для потенциального применения в программных и аппаратных средствах защиты информации.

ЛИТЕРАТУРА

1. *Fomichev V. and Koreneva A.* Encryption performance and security of certain wide block ciphers // J. Comput. Virol. Hack. Tech. 2020. https://doi.org/10.1007/s11416-020-00351-1
2. *Fomichev V. M., Koreneva A. M., Miftahutdinova A. R., and Zadorozhniy D. I.* Evaluation of the maximum performance of block encryption algorithms // Math. Aspects Cryptogr. 2019. V. 10. No. 2. P. 7–16.
3. ISO/IEC 18033-3. IT Security Techniques. Encryption Algorithms. P. 3: Block Ciphers. https://www.iso.org/standard/54531.html.
4. Криптографическая кроссплатформенная C++ библиотека Crypto++ 8.2 с открытым исходным кодом. https://www.cryptopp.com/

УДК 519.17 DOI 10.17223/2226308X/13/19

ХАРАКТЕРИСТИКИ АЛГОРИТМА КОНТРОЛЯ ЦЕЛОСТНОСТИ ДАННЫХ НА ОСНОВЕ АДДИТИВНЫХ ГЕНЕРАТОРОВ И s-БОКСОВ

В. М. Фомичев, А. М. Коренева, Т. Р. Набиев

При проведении анализа программного обеспечения актуальна задача контроля целостности данных больших массивов, при решении которой важно обеспечить приемлемый компромисс между криптографическими свойствами алгоритма контроля целостности и ресурсами, необходимыми для его реализации. Для блоков данных размера 1 кбайт (1024 байта) предложен алгоритм генерации 128-битового кода контроля целостности (ККЦ) с положительными (с позиции синтеза) эксплуатационными и криптографическими свойствами. Алгоритм построен на основе преобразований аддитивных генераторов и s-боксов и реализует функцию $\psi(g^t)\colon V_{2^{13}} \to V_{128}$ со свойством полного перемешивания входных данных. При $6 \leqslant t \leqslant 100$ каждый бит кода существенно зависит от всех битов информационного блока. При случайном равновероятном выборе начального состояния u вероятность получить любой код Q оценивается величиной 2^{-128}. Среднее число опробований пар блоков (u, u'), где $u \neq u'$ и $Q(u) = Q(u')$, приблизительно равно 2^{64}. Сложность вычисления функции $\psi(g^t)$ имеет порядок $t(5u + 8v)$, где u — вычислительная сложность суммирования двух чисел по модулю 2^{64}; v — сложность вычисления s-бокса. В соответствии с проведёнными экспериментами скорость генерации ККЦ варьируется в пределах от 3500 ($t = 6$) до 250 Мбит/с ($t = 96$), соответственно при тех же значениях t время генерации ККЦ варьируется в пределах от 18 до 250 мкс.

Ключевые слова: *аддитивные генераторы, контроль целостности, матрично-графовый подход, перемешивающие свойства, регистры сдвига.*

Введение

Одной из важных задач защиты информации является контроль целостности, который осуществляется с помощью присоединения создателем информации к информа-

ционному l-битовому блоку m-битового кода контроля целостности, $m < l$, представляющего собой двоичную комбинацию, функционально связанную с блоком. Для генерации ККЦ обычно применяются криптографические хэш-функции (SHA, ГОСТ 34.11-2018 и др.) или алгоритмы генерации циклических избыточных кодов (CRC16, CRC32 и др.). Надёжные криптографические хэш-функции требуют значительных ресурсов. При использовании циклических избыточных кодов, обеспечивающих помехоустойчивое кодирование, сложность нахождения коллизии не высока. Поэтому актуально построение альтернативных алгоритмов генерации ККЦ, обладающих следующими положительными свойствами:

— биективность преобразования, на основе которого строится алгоритм генерации ККЦ, это минимизирует вероятность совпадения ККЦ для разных блоков;
— полное перемешивание входных данных (существенная зависимость каждого бита ККЦ от каждого бита блока данных), это затрудняет навязывание ложных блоков и более надёжно обеспечивает целостность данных;
— невысокая вычислительная и емкостная (по памяти) сложность реализации, позволяющая экономить ресурсы при контроле целостности больших массивов данных.

Для повышения надёжности контроля целостности можно дополнительно использовать известные методы [1]: включение в блоки данных меток времени, номеров блоков (или оба приёма одновременно); использование ККЦ, зависящих от секретных параметров (ключей), что сильно усложняет подделку ККЦ.

1. Алгоритм генерации ККЦ

Обозначим: n, m — натуральные числа; V_n — множество двоичных n-мерных векторов; $\mathbb{Z}_{2^n}$ — кольцо вычетов по модулю 2^n; $\overline{X}$ — двоичное представление числа X из кольца $\mathbb{Z}_{2^{64}}$; $\boxplus$ — сложение чисел в кольце $\mathbb{Z}_{2^{64}}$; $\oplus$ — суммирование двоичных строк по модулю 2.

Булева функция называется вполне перемешивающей [2], если она существенно зависит от каждой переменной. Отображение $V_n \to V_m$ называется вполне перемешивающим, если каждая его координатная функция вполне перемешивающая.

Обозначим: $s_0(a_0, \ldots, a_7)$,…,$s_7(a_0, \ldots, a_7)$ — булевы координатные функции вполне перемешивающего преобразования $s(a_0, \ldots, a_7)$ (s-бокса размера 8×8 бит); φ — регистровое преобразование множества состояний аддитивного генератора длины 16 над множеством V_{64} с одной обратной связью $f(X_0, \ldots, X_{15})$, где в ячейке регистра записан вычет $X \in \mathbb{Z}_{2^{64}}$ или, что равносильно, вектор $\overline{X} \in V_{64}$:

$$\varphi(X_0, \ldots, X_{15}) = (X_1, \ldots, X_{15}, X_0 \boxplus X_5 \boxplus X_{10} \boxplus X_{15}).$$

Построим алгоритм генерации r-битового ККЦ для информационного l-битового блока, где $l = 2^{13}$ бит (1 кбайт), $r = 128$. Алгоритм реализует функцию $\psi(g^t)\colon V_{2^{13}} \to V_{128}$, где $g\colon V_{2^{13}} \to V_{2^{13}}$ — преобразование регистрового типа множества состояний схемы из восьми идентичных аддитивных генераторов $AG_0, \ldots, AG_7$, модифицированное с помощью преобразования $s(a_0, \ldots, a_7)$ (рис. 1).

Алгоритм моделируется автономным автоматом Мили без выходов $A = (V_{8,16,64}, g)$, где g — функция переходов и $V_{8,16,64} = \{x_{i,j,k}\}$ — множество состояний автомата, представимое как трёхмерное множество целых неотрицательных чисел, множество координат которых биективно соответствует подмножеству P элементов трёхмерного пространства с целыми координатами, ограниченному параллелепипедом (рис. 2): $0 \leqslant i < 8$, $0 \leqslant j < 16$, $0 \leqslant k < 64$.

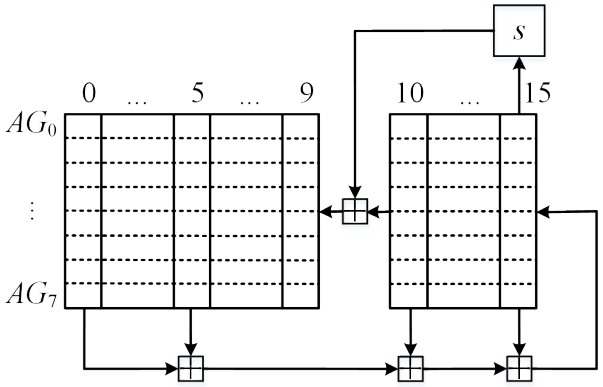

Рис. 1. Регистр над $((\mathbb{Z}_{2^{64}})^8, \boxplus)$

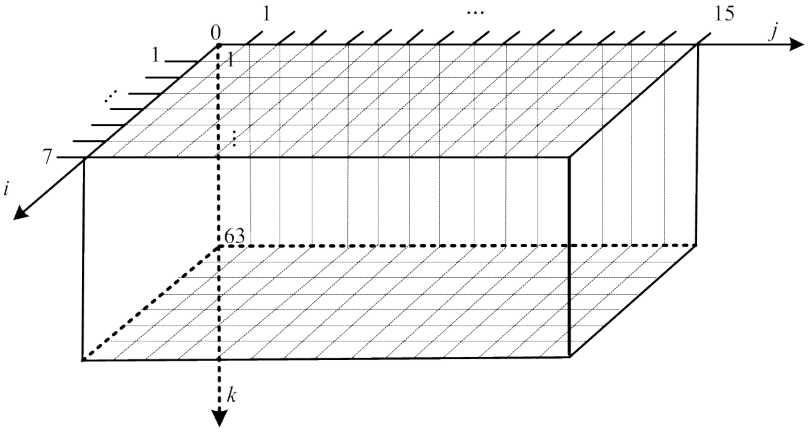

Рис. 2. Параллелепипед, содержащий множество вершин перемешивающего графа преобразования g

Множество состояний автомата в такте $t \geqslant 0$ обозначим $V^{(t)}_{8,16,64} = \{x^{(t)}_{i,j,k}\}$, или матрицей $M^{(t)}_A = \left(X^{(t)}_{i,j}\right)$ над $\mathbb{Z}_{2^{64}}$, где $\overline{X}^{(t)}_{i,j} = (x^{(t)}_{i,j,0}, \ldots, x^{(t)}_{i,j,63})$ — состояние на t-м такте j-й ячейки AG_i.

Построим функцию переходов автомата, используя отображение $z(s)\colon V_8 \to V_{64}$, зависящее от преобразования s, реализуемого s-боксом. При $t \geqslant 0$ определим 8-битовую строку $\omega^{(t)} = (\sigma(\overline{X}^{(t)}_{0,15}, \ldots, \sigma(\overline{X}^{(t)}_{7,15}))$, где $\sigma(x_0, \ldots, x_{63}) = x_0 \oplus \ldots \oplus x_{63}$ — булева функция, определяющая чётность веса вектора $(x_0, \ldots, x_{63})$. Построим 64-битовую конкатенацию $S^{(t)}$ восьми байтов:

$$S^{(t)} = (s^{(t)}_0(\omega^{(t)}) \ldots\ s^{(t)}_7(\omega^{(t)})), \tag{1}$$

где $s^{(t)}_0(\omega^{(t)}) = s(\omega^{(t)})$, $s^{(t)}_j(\omega^{(t)}) = s(s^{(t)}_{j-1}(\omega^{(t)}) \oplus \omega^{(t)})$, $j = 1, \ldots, 7$.

Функция переходов автомата задана равенствами

$$(X^{(t+1)}_{i,0}, \ldots, X^{(t+1)}_{i,15}) = (Y^{(t)}_{i,1}, \ldots Y^{(t)}_{i,15}, f(Y^{(t)}_{i,0}, \ldots Y^{(t)}_{i,15})),\ 0 \leqslant i < 8,$$

где $Y^{(t)}_{i,j} = X^{(t)}_{i,j}$ при $j \neq 10$ и $Y^{(t)}_{i,10} = X^{(t)}_{i,10} \boxplus S^{(t)}$, $S^{(t)}$ вычисляется по формуле (1).

Код Q, генерируемый алгоритмом, определим как 128-битовую строку:

$$Q(V_{8,16,64}) = (X^{(t)}_{0,15} \boxplus X^{(t)}_{1,15} \boxplus X^{(t)}_{2,15} \boxplus X^{(t)}_{3,15}, X^{(t)}_{4,15} \boxplus X^{(t)}_{5,15} \boxplus X^{(t)}_{6,15} \boxplus X^{(t)}_{7,15}). \tag{2}$$

2. Исследование множества образов отображения $z(s)\colon V_8 \to V_{64}$

Вектор $S^{(t)}$ используется в алгоритме как псевдослучайный сдвиг векторов $X_{i,10}^{(t)}$, $0 \leqslant i < 8$. Полагаем, что наилучшие свойства алгоритма генерации ККЦ достигаются, в частности, если для каждого вектора $y \in V_8$ все байты вектора $S^{(t)}(y)$ различны (согласно (1), свойство $S(0,\ldots,0) = (0,\ldots,0)$ должно быть исключено). Отсюда вероятность того, что $S^{(t)}(y)$ содержит повторяющиеся байты, должна быть не больше вероятности 0,1045 этого события для случайного вектора.

Свойство векторов $S^{(t)}$ исследовано с помощью эксперимента на ПЭВМ. При заданном преобразовании s для каждого $y \in V_8$ вычислен вектор $S(y) = (y_0, y_1, \ldots, y_7)$, где $y_0 = s(y)$, $y_j = s(y_{j-1} \oplus y)$, $j = 1,\ldots,7$, и посчитано число различных байтов, составляющих $S(y)$. В табл. 1 приведено число ν_r векторов $S(y)$ (для 256 возможных значений y), состоящих ровно из r различных байтов среди $y_0,\ldots,y_7$, $r = 1,\ldots,8$; результаты получены для s-боксов размера 8×8 известных блочных шифров.

Таблица 1

Число ν_r векторов $S(y)$, состоящих из r различных байтов

S-бокс	Значение r							
	1	2	3	4	5	6	7	8
Кузнечик	1	0	0	0	0	0	2	253
AES	1	2	2	1	0	0	1	249
AES_inv	1	0	1	1	1	1	1	250
SM4	1	1	1	0	5	0	1	247
CRYPTON S_0	1	0	1	1	2	2	1	248
CRYPTON S_1	1	0	3	0	0	1	2	249
CRYPTON S_2	1	1	1	0	1	1	1	250
CRYPTON S_3	1	1	2	1	0	1	0	250
Camellia	1	2	0	0	1	1	0	251
KHAZAD-0	1	1	1	3	0	1	0	249
KHAZAD	1	3	0	1	0	2	1	248
CLEFIA S_0	1	1	1	0	1	1	2	249
CLEFIA S_1	1	1	0	0	2	1	1	250
Kalyna π_0	1	1	1	2	2	0	0	249
Kalyna π_1	1	1	0	0	2	2	0	250
Kalyna π_2	1	1	0	3	1	0	1	249
Kalyna π_3	1	1	0	1	2	1	2	248

При использовании s-боксов табл. 1 вероятность того, что в последовательности $S^{(t)}(y)$ есть повторяющиеся байты, не больше 0,0351 (s-бокс SM4). Вероятность такого события наименьшая (0,0117) при использовании s-бокса алгоритма «Кузнечик».

3. Характеристики алгоритма генерации ККЦ

1. Преобразование g биективное. Число прообразов любого значения функции $\psi(g^t)$ равно 2^{l-2r}. Следовательно, при случайном равновероятном выборе начального состояния u из множества $V_{8,16,64}$ вероятность получить заданный код Q равна 2^{-128}. Среднее число опробований для поиска пар блоков (u, u'), таких, что $u \neq u'$ и $Q(u) = Q(u')$, оценивается с помощью парадокса дней рождения величиной порядка 2^{64}.

2. Перемешивающие свойства алгоритма оценены с помощью развития матрично-графового подхода, применённого в [3] для оценки перемешивающих свойств преоб-

разований модифицированных аддитивных генераторов (АГ). Для свойства полного перемешивания координатных функций, соответствующих крайним ячейкам АГ (это свойство необходимо в соответствии с формулой (2)), оценен локальный экспонент перемешивающего орграфа преобразования g. Оценка, равная 6, получена как длина путей из одной вершины в другую для всех допустимых пар вершин вида $((i, 15, j), (i', 15, j'))$ [4, с. 457], проходящих через некоторую вершину с петлей. Для контроля целостности необходимо, чтобы ККЦ вычислялся с помощью вполне перемешивающей функции. Установлено, что при $t \geqslant 6$ обе формирующие код Q функции $X_{0,15}^{(t)} \boxplus X_{1,15}^{(t)} \boxplus X_{2,15}^{(t)} \boxplus X_{3,15}^{(t)}$ и $X_{4,15}^{(t)} \boxplus X_{5,15}^{(t)} \boxplus X_{6,15}^{(t)} \boxplus X_{7,15}^{(t)}$ являются вполне перемешивающими. Экспериментально определено, что свойство полного перемешивания этих функций сохраняется при $6 \leqslant t \leqslant 100$.

3. Сложность вычисления функции $\psi(g^t)$ оценивается величиной порядка $t(5u + 8v)$, где u — вычислительная сложность суммирования двух чисел по модулю 2^{64}; v — сложность вычисления s-бокса. В табл. 2 даны результаты измерения скорости генерации и времени вычисления 128-битового ККЦ для блока данных размера 1 кбайт при различных t. Эксперименты проведены на ПЭВМ с процессором Intel Core i5-8600 и тактовой частотой 3,1 ГГц.

Таблица 2

Скорость генерации и время вычисления ККЦ

Число тактов, t	6	12	18	36	72	96
Скорость генерации, Мбит/с	3500	1900	1200	650	330	250
Время вычисления, мкс	18	32	49	96	200	250

Выводы

Предложен новый класс алгоритмов на основе функций аддитивных генераторов и s-боксов для генерации кодов контроля целостности блоков данных объёма 1 кбайт. Подход может быть распространён на блоки данных большего объёма. Алгоритмы обладают положительными эксплуатационными и криптографическими свойствами: невысокой сложностью реализации и свойством полного перемешивания входных данных, что существенно затрудняет применение ряда методов криптоанализа.

ЛИТЕРАТУРА

1. *Будзко В. И., Мельников Д. А., Фомичёв В. М.* Базовые требования к подсистемам обеспечения криптоключами в информационно-технологических системах высокой доступности // Системы высокой доступности. 2016. Т. 12. № 3. С. 73–82.
2. *Fomichev V. M.* Matrix-graph approach for studying nonlinearity of transformations on vector space // VIII симп. «Современные тенденции в криптографии» CTCrypt 2019. `https://ctcrypt.ru/files/files/2019/materials/08_Fomichev.pdf`
3. *Fomichev V. M. and Koreneva A. M.* Mixing properties of modified additive generators // J. Appl. Ind. Math. 2017. V. 11. P. 215–226.
4. *Fomichev V. M., Avezova Ya. E., Koreneva A. M., and Kyazhin S. N.* Primitivity and local primitivity of digraphs and nonnegative matrices // J. Appl. Ind. Math. 2018. V. 12. P. 453–469.

УДК 004.056.55 DOI 10.17223/2226308X/13/20

АНАЛИЗ РЕЖИМОВ ШИФРОВАНИЯ ДЛЯ РЕАЛИЗАЦИИ В УСТРОЙСТВАХ RFID

К. Д. Царегородцев

Технология радиочастотной идентификации (RFID) описывает способы бесконтактной идентификации и аутентификации объектов с возможным обменом зашифрованными данными. В состав RFID-системы входит радио-метка и считывающее устройство. Основная функция RFID-меток — аутентификация с передачей небольшого объёма информации между меткой и считывателем (например, платежи). С учётом аппаратных ограничений изучены режимы шифрования, для которых на метке необходимо реализовать только алгоритм зашифрования блока текста. Рассмотрены режимы CTR, OFB, CFB, модифицированный режим CBC. Для модифицированного режима CBC получена верхняя оценка стойкости в соответствующей модели противника.

Ключевые слова: *доказуемая стойкость, RFID, режим шифрования.*

RFID-система состоит из двух взаимодействующих участников: метки (с записанными на ней наборами ключей) и считывателя. Необходимо передавать зашифрованную информацию как от считывателя к метке, так и от метки к считывателю. Дополнительно налагаются ограничения на метку: на ней реализован только алгоритм зашифрования блока текста (без расшифрования).

Будем рассматривать следующую стандартную модель LOR-неразличимости двух режимов шифрования [1]. На каждом шаге вычислений противник (являющийся вероятностной машиной Тьюринга) подаёт запросы на вход одного из двух оракулов: $\mathcal{O}_1$ или $\mathcal{O}_2$. Первый оракул реализует режим шифрования, используемый при передаче информации от метки к считывателю, второй оракул — режим шифрования, используемый при передаче информации от считывателя к метке. Оракулы используют один и тот же ключ, выбранный случайно равновероятно в начале эксперимента. Тем самым оракулы связаны друг с другом посредством общего ключа.

На каждом шаге противник даёт одному из оракулов (любому, на его выбор) два сообщения (одинаковой длины) для обработки:

$$\left(M^L = (m_1^L, \ldots, m_t^L),\ M^R = (m_1^R, \ldots, m_t^R)\right).$$

В эксперименте Left каждый из оракулов $\mathcal{O}_1$ и $\mathcal{O}_2$ зашифровывает сообщение M^L в своём режиме шифрования и возвращает вычисленный шифртекст. В эксперименте Right каждый из оракулов зашифровывает сообщение M^R и возвращает вычисленный шифртекст.

Задача противника — анализируя полученные шифртексты, суметь различить два эксперимента. Если противник «думает», что оракулы зашифровывают правые тексты (эксперимент Right), то он выдаёт 1, в противном случае — 0. Если противник способен с высокой вероятностью различать эксперименты Left и Right, то это означает, что он может восстанавливать частичную информацию об открытом тексте из шифртекста. Таким образом, преимущество противника $\mathcal{A}$ задаётся как разность вероятностей

$$\mathrm{Adv}(\mathcal{A}) = \mathsf{P}[\mathrm{Right}(\mathcal{A}) \to 1] - \mathsf{P}[\mathrm{Left}(\mathcal{A}) \to 1],$$

где $\mathsf{P}[X(\mathcal{A}) \to 1]$ — вероятность того, что противник, взаимодействуя с экспериментатором X, выдаст 1. Вероятность берётся по начальному выбору ключа, внутренних

выборах оракулов (случайные векторы инициализации) и случайным битам самого противника. Так, например, если противник не делает никаких запросов к оракулам и просто выдаёт результат подбрасывания случайной равновероятной монеты, то его преимущество $\mathrm{Adv}(\mathcal{A})$ равно нулю. Если противник идеально различает два эксперимента, то его преимущество равно 1.

Обозначим: q — общее число запросов к оракулам $\mathcal{O}_1$ и $\mathcal{O}_2$; m — максимальная длина одного запроса (в блоках); t — количество тактов вычислений противника; n — длина одного блока (в битах); k — длина используемого ключа; $\mathrm{Adv}^{LOR}_{\mathcal{O}_1,\mathcal{O}_2}(t,q,m)$ — максимально достижимое преимущество среди всех противников, работающих за время не более t и делающих суммарно не более q запросов, каждый из которых имеет длину не более m блоков.

Режимы шифрования CTR (гаммирования), OFB (гаммирования с обратной связью по выходу), CFB (гаммирования с обратной связью по шифртексту) [2] не требуют реализации алгоритма расшифрования на метке, поэтому могут использоваться непосредственно, без модификаций. Для этих режимов оракулы $\mathcal{O}_1$ и $\mathcal{O}_2$ совпадают.

Теорема 1 [3]. Для режима CTR имеем следующую оценку:

$$\mathrm{Adv}^{\mathrm{LOR}}_{\mathrm{CTR},\,\mathrm{CTR}}(t,q,m) \leqslant \frac{2q^2m^2}{2^n} + \frac{t+q+bqm}{2^{k-1}}.$$

Аналогичные оценки могут быть получены для режимов OFB и CFB (главный член имеет порядок $O((qm)^2/2^n)$ — оценка дней рождения [4]). При доказательстве используется предположение о PRP-стойкости используемого блочного шифра.

Отдельно рассмотрим режим CBC. Для расшифрования в режиме CBC требуется реализация алгоритма расшифрования блока текста, поэтому режим CBC при передаче сообщения от считывателя к метке был модифицирован следующим образом (обозначен далее как $\widehat{\mathrm{CBC}}$):

$$C_0 = \mathrm{IV}, \quad C_i = E_k^{-1}(C_{i-1} \oplus M_i),$$

где E_k^{-1} — алгоритм расшифрования блока текста; M_i — i-й блок открытого текста; C_i — i-й блок шифртекста. Заметим, что для расшифрования сообщения, зашифрованного по алгоритму $\widehat{\mathrm{CBC}}$, не требуется реализации алгоритма расшифрования блока текста.

Таким образом, у противника есть доступ к оракулу зашифрования по алгоритму CBC (оракул $\mathcal{O}_1$) и по алгоритму $\widehat{\mathrm{CBC}}$ (оракул $\mathcal{O}_2$) на одном и том же ключе k.

Основным результатом является следующая

Теорема 2. Для пары режимов CBC и $\widehat{\mathrm{CBC}}$ выполнена оценка

$$\mathrm{Adv}^{\mathrm{LOR}}_{\mathrm{CBC},\widehat{\mathrm{CBC}}}(t,q,m) \leqslant \frac{3q^2m^2}{2^n - qm} + \frac{t+q}{2^{k-1}}.$$

Доказательство основано на идее из работы [5]: если выходы двух оракулов могут быть промоделированы генератором, которому на вход подаются лишь длины сообщений (но не их содержание), то режим является LOR-стойким.

На первом шаге доказательства, используя свойство sPRP-стойкости блочного шифра (стандартное предположение, см., например, [6]), мы заменяем каждое вхождение блочного шифрования $E_k(x)$ и $E_k^{-1}(x)$ на применение случайной подстановки $\pi(x)$ и $\pi^{-1}(x)$ соответственно.

На втором шаге анализируем полученную конструкцию и показываем, что выходы обоих оракулов можно промоделировать без знания открытого текста. Итоговые оценки получаются из предположения, что множества, на которых вычисляются значения подстановок $\pi(x)$ и $\pi^{-1}(x)$, не пересекаются.

Полученная оценка близка к оптимальной для стандартного режима CBC: член вида $O((qm)^2/2^n)$ отражает тот факт, что для режима CBC всегда существует атака дней рождения, предполагающая возникновение коллизии для векторов инициализации IV.

ЛИТЕРАТУРА

1. *Katz J. and Lindell Y.* Introduction to Modern Cryptography, 2nd Ed. Chapman & Hall/CRC, 2014.
2. Межгосударственный стандарт ГОСТ 34.13-2018 Информационная технология (ИТ). Криптографическая защита информации. Режимы работы блочных шифров. М.: Стандартинформ, 2018.
3. *Ahmetzyanova L. R., Alekseev E. K., Oshkin I. B., et al.* On the properties of the CTR encryption mode of Magma and Kuznyechik block ciphers with re-keying method based on CryptoPro Key Meshing // Матем. вопр. криптогр. 2017. Т. 8. № 2. С. 39–50.
4. *Rogaway P.* Evaluation of Some Block Cipher Modes of Operation. 2011. `https://web.cs.ucdavis.edu/~rogaway/papers/modes.pdf`
5. *Wooding M.* New Proofs for Old Modes. IACR Cryptology ePrint Archive. 2008.
6. *Bellare M., Desai A., Jokipii E., and Rogaway P.* A concrete security treatment of symmetric encryption // Proc. 38th Ann. Symp. Foundations of Computer Science, IEEE, 1997. P. 394–403.

УДК 519.714.5 DOI 10.17223/2226308X/13/21

ОБ ОДНОМ ПОДХОДЕ К ПОСТРОЕНИЮ КРАТНО ТРАНЗИТИВНОГО МНОЖЕСТВА БЛОЧНЫХ ПРЕОБРАЗОВАНИЙ

И. В. Чередник

Пусть Ω — произвольное конечное множество; $\mathcal{B}(\Omega)$ — семейство всех бинарных операций, определённых на Ω; $x_1, \dots, x_n$ — переменные, принимающие значения из Ω; $*_1, \dots, *_k$ — общие символы бинарных операций. Фиксированный набор $W = (w_1, \dots, w_m)$ формул в алфавите $\{x_1, \dots, x_n, *_1, \dots, *_k\}$ при замене $*_1, \dots, *_k$ на произвольные бинарные операции $F_1, \dots, F_k \in \mathcal{B}(\Omega)$ соответственно реализует отображение $W^{F_1,\dots,F_k} \colon \Omega^n \to \Omega^m$. Исследованы криптографические свойства (биективность и кратная транзитивность) семейств блочных преобразований $\{W^{F_1,\dots,F_k} : F_1, \dots, F_k \in \mathcal{K}\}$, $\mathcal{K} \subset \mathcal{B}(\Omega)$, которые могут быть использованы при построении хэш-функций и блочных шифров.

Ключевые слова: *блочные преобразования, кратная транзитивность множества блочных преобразований, функциональная бинарная сеть.*

В последнее время при разработке систем защиты информации активно исследуется возможность использования неассоциативных алгебраических структур, особое место в таких исследованиях занимают квазигруппы. Например, в ряде схем поточных шифров, хеш-функций и др. [1 – 3] используются семейства блочных преобразований, реализуемых наборами «цепных» формул вида

$$C_a^*(x_1, \dots, x_n) = (a * x_1, (a * x_1) * x_2, \dots, ((a * x_1) * \dots) * x_n), \quad a \in \Omega,$$

где $*$ — квазигрупповая операция на некотором конечном множестве Ω. При этом в подавляющем большинстве схем подобного рода квазигрупповая операция $*$ выбирается из небольшого множества отобранных квазигрупп $\{*_1, \ldots, *_k\}$ и параметризация соответствующих блочных преобразований $C_a^*\colon \Omega^n \to \Omega^n$ достигается в основном за счёт выбора «начального» элемента $a \in \Omega$.

Одной из желаемых характеристик семейств блочных преобразований, используемых в узлах защиты информации, является кратная транзитивность данного семейства. Однако неизвестно, являются ли классы блочных преобразований типа

$$\left\{C_{a_1}^{*_{i_1}} \cdot \ldots \cdot C_{a_r}^{*_{i_r}} :\ a_1, \ldots, a_r \in \Omega,\ i_1, \ldots, i_r \in \{1, \ldots, k\}\right\}$$

хотя бы транзитивными, при этом отсутствуют практически эффективные методы, которые позволяли бы это выяснить.

В данной работе предлагается концепция построения классов блочных преобразований, при которой параметризация отображений достигается исключительно за счет широкого выбора бинарных операций. Пусть Ω — произвольное конечное множество; $\mathcal{B}(\Omega)$ — множество всех бинарных операций, определённых на Ω; $\{x_1, \ldots, x_n\}$ — множество переменных и $*_1, \ldots, *_k$ — символы бинарных операций. Произвольная формула $w(x_1, \ldots, x_n)$ в алфавите $\{x_1, \ldots, x_n, *_1, \ldots, *_k\}$ при сопоставлении символам $*_1, \ldots, *_k$ конкретных бинарных операций $F_1, \ldots, F_k \in \mathcal{B}(\Omega)$ соответственно реализует функцию $w^{F_1,\ldots,F_k}\colon \Omega^n \to \Omega$, а набор формул $W = (w_1, \ldots, w_m)$ — отображение $W^{F_1,\ldots,F_k}\colon \Omega^n \to \Omega^m$.

Предложенная концепция во многом происходит от практики, поскольку при проведении анализа узлов переработки информации часто возникает задача исследования семейств отображений вида

$$\{W^{F_1,\ldots,F_k} : F_1, \ldots, F_k \in \mathcal{K}\}, \quad \mathcal{K} \subset \mathcal{B}(\Omega). \tag{1}$$

Так, например, в некоторых случаях наличие запретов в совместных распределениях нескольких отображений из класса (1) позволяет идентифицировать начальные состояния и часть постоянных параметров изучаемых узлов.

Отметим также, что изложенная концепция в случае использования одной бинарной операции уже рассматривалась в работах [4–9]. Так, в работе [5] для некоторых семейств преобразований

$$\{W^F : F \in \mathcal{Q}(\Omega)\} \tag{2}$$

($\mathcal{Q}(\Omega)$ — множество всех бинарных квазигрупп, заданных на Ω) предложена модель наглядного описания в виде бинарной функциональной схемы-сети. Указанное представление позволило в работах [5, 7] строго описать и обосновать методы исследования кратной транзитивности классов преобразований вида (2). Кроме того, в [5, 7] изложены алгоритмы построения семейств вида (2) с требуемой кратной транзитивностью. Однако в большинстве узлов защиты информации вовсе не требуется использование квазигруппы, и достаточно бинарной операции, обратимой по одной, например правой, переменной. Поэтому в [8, 9] исследована возможность продолжения результатов работ [5, 7] на более широкий по сравнению с $\mathcal{Q}(\Omega)$ класс $\mathcal{B}^*(\Omega)$ всех бинарных операций, обратимых по правой переменной.

В данной работе получено следующее продолжение результатов [5, 7–9]:

1) предложенная в [5] модель наглядного представления семейств преобразований типа (2) в виде бинарной функциональной схемы-сети пригодна также

для представления произвольных семейств преобразований вида (1) и позволяет строго описать методы исследования кратной транзитивности произвольных семейств преобразований вида (1) для любого класса $\mathcal{K}$, удовлетворяющего условию $\mathcal{Q}(\Omega) \subset \mathcal{K} \subset \mathcal{B}^*(\Omega)$;

2) все основные результаты работ [5, 7 – 9] корректным образом распространяются на случай использования нескольких бинарных операций — такой более общий подход улучшает характеристики практического использования кратно транзитивных семейств преобразований, предложенных в [7, 9], а кроме того, позволяет «аппроксимировать» некоторые известные блочные шифры, в которых S-боксы зависят от ключа, (Blowfish, Twofish и др.) семействами блочных преобразований вида (1), и, как следствие, появляется возможность оценить кратную транзитивность указанных блочных шифров.

ЛИТЕРАТУРА

1. *Gligoroski D., Markovski S., Kocarev L., and Gusev M.* Edon80. `http://www.ecrypt.eu.org/stream/edon80p3.html` — eSTREAM, ECRYPT Stream Cipher Project.
2. *Gligoroski D., Markovski S., and Kocarev L.* Edon-R, An infinite family of cryptographic hash functions. `http://csrc.nist.gov/pki/HashWorkshop/2006/Papers/GLIGOROSKI_EdonR-ver06.pdf` — Second NIST Cryptographic Hash Workshop.
3. *Gligoroski D., Markovski S., and Knapskog S.* A public key block cipher based on multivariate quadratic quasigroups. `http://eprint.iacr.org/2008/320` — Cryptology ePrint Archive.
4. *Чередник И. В.* Об одном подходе к построению транзитивного множества блочных преобразований // Прикладная дискретная математика. Приложение. 2017. № 10. С. 27–29.
5. *Чередник И. В.* Один подход к построению транзитивного множества блочных преобразований // Прикладная дискретная математика. 2017. № 38. С. 5–34.
6. *Чередник И. В.* k-Транзитивность одного класса блочных преобразований // Прикладная дискретная математика. Приложение. 2018. № 11. С. 21–23.
7. *Чередник И. В.* Один подход к построению кратно транзитивного множества блочных преобразований // Прикладная дискретная математика. 2018. № 42. С. 18–47.
8. *Чередник И. В.* Об использовании бинарных операций при построении транзитивного множества блочных преобразований // Дискретная математика. 2019. № 31. Т. 3 С. 93–113.
9. *Чередник И. В.* Об использовании бинарных операций при построении кратно транзитивного множества блочных преобразований // Дискретная математика. 2020. Т. 32. № 2. С. 85–111.

УДК 519.719.1 DOI 10.17223/2226308X/13/22

УТОЧНЕНИЕ СТРАТЕГИИ МАЙНИНГА ДЛЯ НЕБОЛЬШОЙ ГРУППЫ УЧАСТНИКОВ

А. В. Черемушкин

Ittay Eyal и Emin Gün Sirer описали стратегию проведения т. н. корыстного майнинга, показывающую уязвимость протокола формирования цепочки блоков, реализованного в биткоине, к атаке со стороны группы участников майнинга, составляющей относительно небольшую часть от общего числа майнеров, и позволяющую ей получить вознаграждение, превышающее размер доли имеющихся у них вычислительных ресурсов. В настоящей работе предложена уточнённая вероятностно-автоматная марковская модель, основанная на предположении о независимости обеих групп участников.

Ключевые слова: *блокчейн, майнинг, марковская модель, вероятностный автомат.*

В работе приводится уточнение вероятностной модели стратегии поведения выделенной (корыстной) группы участников майнинга, у которой суммарная вычислительная мощность принадлежащих им ресурсов не превосходит половины от общей вычислительной мощности [1–3]. Пусть доля вычислительных ресурсов корыстной группы пропорциональна $p = \alpha < 1/2$, а у второй группы, составленной из остальных участников, $q = 1 - \alpha$. Авторы [1–3] исходят из предположения, что поведение групп участников моделируется биномиальным распределением с вероятностями p успешного подбора корыстной группой участников и q для случая успешного подбора группой остальных участников.

В отличие от [3], будем предполагать, что обе группы участников действуют независимо, поэтому переходы между состояниями определяются не одной случайной величиной, а двумя независимыми случайными величинами ξ_1 и ξ_2 с вероятностями успеха $\mathsf{P}[\xi_1 = 1] = p$ (для первой группы) и $\mathsf{P}[\xi_2 = 1] = q$ (для второй группы) соответственно. При этом возможны не только ситуации, когда успех имеется у одной из сторон, но и ситуации, когда обе стороны одновременно добиваются успеха, а также когда успеха не добивается ни она из сторон.

Общая идея стратегии корыстного майнинга состоит в том, что в случае успешного подбора цепочки из s блоков корыстная группа не обнародует результат, а держит его в тайне от остальных до тех пор, пока остальные участники сами не подберут очередной блок. В этом случае они поступают одним из следующих вариантов:

— если $s = 1$, то они обнародуют свой блок, создавая разветвление длины 1 и откладывая вопрос о том, какая из групп получит вознаграждение;
— если $s = 2$, то корыстная группа раскрывает оба своих блока, тем самым получая вознаграждение за два блока и лишая вознаграждения группу остальных участников;
— если $s \geqslant 3$, то они обнародуют блок, стоящий в начале своей сохраняемой в тайне цепочки, создавая разветвление из двух цепочек, либо увеличивая на 1 длину цепочки в существующем разветвлении, тем самым лишая группу остальных участников выигрыша.

Такая стратегия моделируется с помощью автономного вероятностного автомата, множество состояний которого состоит из трёх групп. Первую группу составляют состояния s_i, $i = 0, 1, 2, \ldots$, в которых у корыстной группы участников имеется преимущество в числе подобранных хеш-значений для блоков, равное номеру состояния. Отрицательные значения не рассматриваются, так как они соответствуют нулевому состоянию. Вторую группу составляют состояния $s_{i,0}$, $i \geqslant 2$, в которых блокчейн допускает разветвление с двумя продолжениями, у которых длина цепочки, сформированной корыстной группой, содержит на i блоков больше, чем цепочка, сформированная группой остальных участников майнинга. Случай $i = 1$ также не рассматривается, так как в этом случае первая группа раскрывает свою цепочку и система переходит в нулевое состояние. Третью группу образуют состояния $s_{i,i}$ при $i \geqslant 1$, которые соответствуют случаю разветвлений с двумя одинаковыми длинами продолжений исходной цепочки.

Авторы [1] рассмотрели также случай, когда при наличии разветвления среди остальных участников найдётся подгруппа, составляющая (по мощности вычислительных ресурсов) долю, равную γ, $0 \leqslant \gamma \leqslant 1$, которая будет пытаться продолжить ветку, созданную выделенной группой, тем самым повышая вероятность получения возна-

граждения корыстной группой за блоки, подобранные ею ранее. Модель [1] позволяет успешно рассчитать вероятность получения вознаграждения, превышающего долю имеющихся у группы вычислительных ресурсов — корыстная группа получает преимущество при выполнении неравенства

$$\frac{1-\gamma}{3-2\gamma} < \alpha < \frac{1}{2}.$$

Для анализа этой ситуации будем, как и раньше, рассматривать вероятностную модель, включающую две группы участников, осуществляющих майнинг с вероятностями успеха p и q. В тех случаях, когда для группы остальных участников имеется выбор того, для какой из цепочек строить продолжение, будем предполагать, что вероятность успешного подбора продолжения для цепочки, содержащей блоки, найденные группой корыстных участников, равна γq, а для второй цепочки в разветвлении блокчейна она равна $(1-\gamma)q$. Поэтому для тех состояний, которые соответствуют разветвлению блокчейна, должно быть не четыре, а шесть вариантов перехода в другие состояния: (два варианта для корыстной группы) $\times$ (три варианта для группы остальных участников). Это моделируется случайной величиной, принимающей три значения 0, 1, 2 с вероятностями p, $q\gamma$, $q(1-\gamma)$ соответственно.

Граф переходов вероятностного автомата, моделирующего поведение двух групп участников, приведён на рис. 1. Вершины графа переходов помечены индексами соответствующих состояний, а переходы — парами ab, соответствующими значениям случайных величин $\xi_1 = a$ и $\xi_2 = b$ $(a, b \in \{0, 1, 2\})$. Из состояний первой группы имеются только четыре возможных перехода (табл. 1), а для состояний второй и третьей групп — шесть (табл. 2). Для изображения рёбер используются линии трёх типов: жирной линией нарисованы рёбра, соответствующие событиям, в которых корыстная группа гарантирует для себя вознаграждение, пунктиром — в которых вознаграждение получает группа остальных участников, а тонкие линии указывают, что ни одна из групп ничего не получает.

Таблица 1

Переходы из состояний первой группы

Метка	Вероятность перехода	Событие
00	qp	Ни одна группа не нашла продолжения
01	q^2	Вторая группа нашла продолжение второй цепочки
10	p^2	Корыстная группа нашла продолжение своей цепочки
11	pq	Обе группы нашли продолжение для своих цепочек

Таблица 2

Переходы из состояний второй и третьей групп

Метка	Вероятность перехода	Событие
00	qp	Ни одна группа не нашла продолжения
01	$q^2\gamma$	Вторая группа нашла продолжение первой цепочки
02	$q^2(1-\gamma)$	Вторая группа нашла продолжение второй цепочки
10	p^2	Корыстная группа нашла продолжение своей цепочки
11	$pq\gamma$	Обе группы нашли продолжение первой цепочки
12	$pq(1-\gamma)$	Обе группы нашли продолжение для своих цепочек

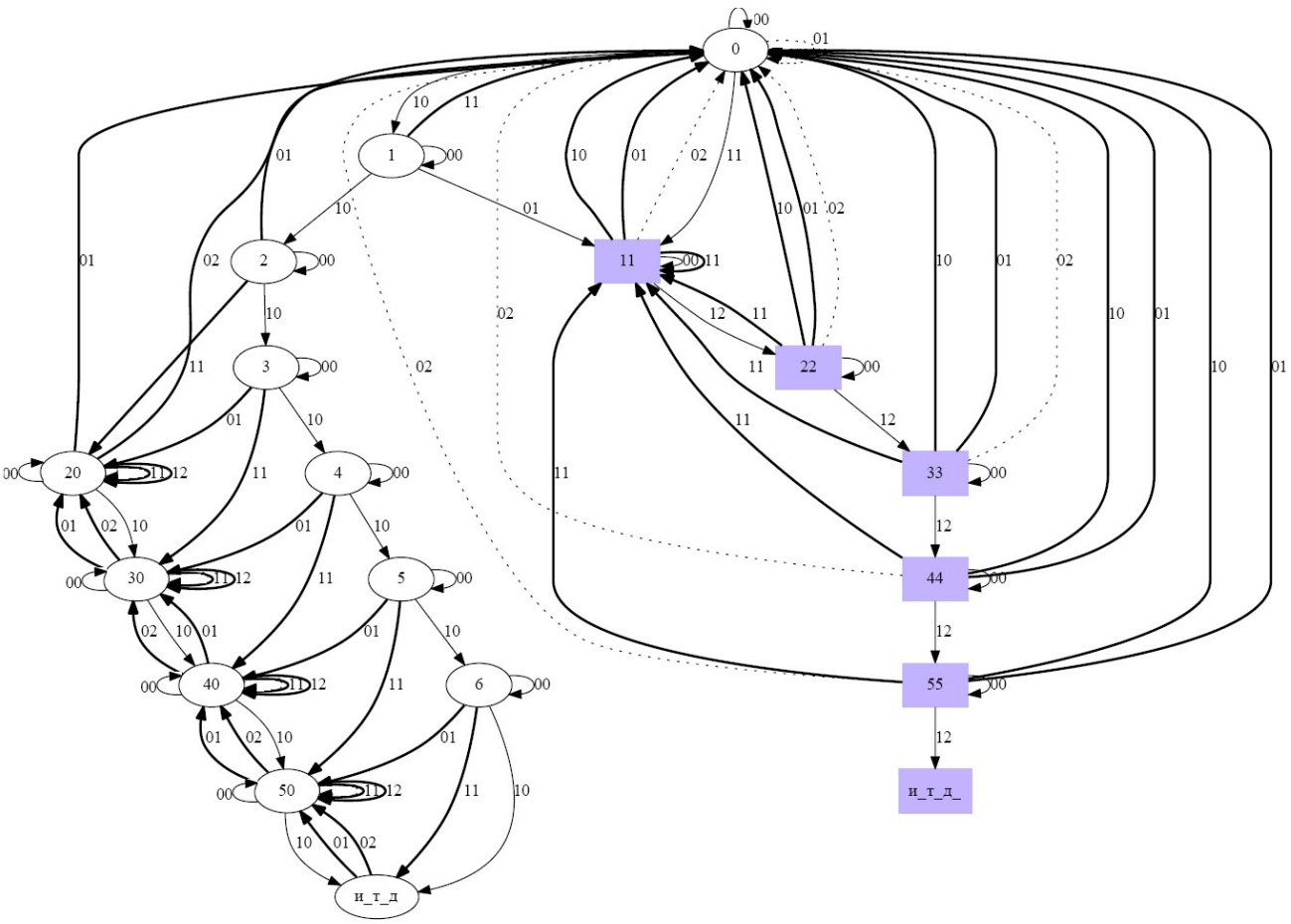

Рис. 1. Граф переходов состояний при $0 \leqslant \gamma \leqslant 1$

Для удобства соберём в одну таблицу возможные переходы автомата (табл. 3).

Для оценки вероятностей выигрыша каждой из групп необходимо сначала вычислить вероятности p_i $(i = 0, 1, \ldots)$, $p_{i,0}$ $(i = 2, 3, \ldots)$ и $p_{i,i}$ $(i = 1, 2, \ldots)$ нахождения системы в каждом из состояний. Будем исходить из предположения, что соответствующая цепь Маркова является стационарной, т. е. эти вероятности не зависят от момента времени. Поэтому вероятности нахождения системы в каждом из состояний должны удовлетворять следующей системе уравнений:

$$
\begin{gathered}
p_0 = (p^2 + q^2) \sum_{i \geqslant 1} p_{i,i} + qp_0 + pqp_1 + q^2 \left(p_2 + p_{2,0}\right), \\
p_i = pq\, p_i + p^2 p_{i-1}, \quad i \geqslant 1, \\
p_{1,1} = qp\, p_{1,1} + pq\, p_0 + q^2 p_1 + pq\gamma \sum_{j \geqslant 1} p_{j,j}, \\
p_{i,i} = pq\, p_{i,i} + pq(1-\gamma)\, p_{i-1,i-1}, \quad i \geqslant 2, \\
p_{2,0} = qp\, p_{2,0} + q^2 p_{3,0} + pq\, p_{2,0} + pq\, p_2 + q^2 p_3, \\
p_{i,0} = qp\, p_{i,0} + p^2 p_{i-1,0} + q^2 p_{i+1,0} + pq\, p_{i,0} + pq\, p_i + q^2 p_{i+1}, \quad i \geqslant 3, \\
\sum_{i \geqslant 0} p_i + \sum_{i \geqslant 1} p_{i,i} + \sum_{i \geqslant 2} p_{i,0} = 1.
\end{gathered}
\tag{1}
$$

Найдём выражения для всех вероятностей через вероятность p_1 и значения параметров p, q и γ.

Таблица 3

Переходы модифицированного графа

Исходное состояние	Метка ребра	Верояность перехода	Следующее состояние	Выигрыш обеих групп
s_i $(i \geqslant 0)$	00	qp	s_i	$(0,0)$
s_0	01	q^2	s_0	$(0,1)$
s_1	01	q^2	$s_{1,1}$	$(0,0)$
s_2	01	q^2	s_0	$(2,0)$
s_i $(i \geqslant 3)$	01	q^2	$s_{i-1,0}$	$(1,0)$
s_i $(i \geqslant 0)$	10	p^2	s_{i+1}	$(0,0)$
s_0	11	pq	$s_{1,1}$	$(0,0)$
s_1	11	pq	s_0	$(2,0)$
s_i $(i \geqslant 2)$	11	pq	$s_{i,0}$	$(1,0)$
$s_{i,i}$	00	pq	$s_{i,i}$	$(0,0)$
$s_{i,i}$	01	$q^2\gamma$	s_0	$(i,1)$
$s_{i,i}$	02	$q^2(1-\gamma)$	s_0	$(0,i+1)$
$s_{i,i}$	10	p^2	s_0	$(i+1,0)$
$s_{i,i}$	11	$pq\gamma$	$s_{1,1}$	$(i,0)$
$s_{i,i}$	12	$pq(1-\gamma)$	$s_{i+1,i+1}$	$(0,0)$
$s_{i,0}$	00	qp	$s_{i,0}$	$(0,0)$
$s_{2,0}$	01	$q^2\gamma$	s_0	$(2,0)$
$s_{i,0}$ $(i>2)$	01	$q^2\gamma$	$s_{i-1,0}$	$(1,0)$
$s_{2,0}$	02	$q^2(1-\gamma)$	s_0	$(2,0)$
$s_{i,0}$ $(i>2)$	02	$q^2(1-\gamma)$	$s_{i-1,0}$	$(1,0)$
$s_{i,0}$	10	p^2	$s_{i+1,0}$	$(0,0)$
$s_{i,0}$	11	$pq\gamma$	$s_{i,0}$	$(1,0)$
$s_{i,0}$	12	$pq(1-\gamma)$	$s_{i,0}$	$(1,0)$

Утверждение 1.

1) Вероятности p_i при $i \geqslant 0$ удовлетворяют соотношению

$$p_{i+1} = \frac{p^2}{1-pq} p_i.$$

2) Вероятности $p_{i,i}$ при $i \geqslant 1$ удовлетворяют соотношениям

$$p_{1,1} = p_1 \frac{q(1-pq(2-\gamma))}{p(1-pq)(1-2pq)}, \qquad p_{i+1,i+1} = \frac{pq(1-\gamma)}{1-pq} p_{i,i}.$$

3) Вероятности $p_{i,0}$ при $i \geqslant 2$ вычисляются по формулам

$$p_{2,0} = p_1 \frac{p^3}{q^2(1-pq)}, \qquad p_{i+1,0} = p_1 \left(\frac{p^2}{q^2}\right)^i \left(\frac{p+q^2}{1-pq} - \left(\frac{q^2}{1-pq}\right)^i\right).$$

Оценим величину $R = r_0/(r_0+r_1)$ доли корыстной группы в общей сумме вознаграждения, полученной при применении описанной стратегии майнинга. Вознаграждение первой группы в этом случае определяется как

$$r_0 = p_1(2pq + q^2 \frac{p^2}{1-pq} + q^2 p_{2,0}) + q \sum_{i \geqslant 2} p_i + q \sum_{i \geqslant 2} p_{i,0} + p^2 \sum_{i \geqslant 1} (i+1) p_{ii} + q\gamma \sum_{i \geqslant 1} i p_{i,i}.$$

Для второй группы вознаграждение равно

$$r_1 = q^2(p_0 + (1-\gamma)p_1\Sigma_1 + \gamma \sum_{i \geqslant 1} p_{i,i}) = q^2 \left(p_1 \frac{1-pq}{p^2} + (1-\gamma) \sum_{i \geqslant 1} (i+1) p_{ii} + \gamma \sum_{i \geqslant 1} p_{i,i}\right).$$

Утверждение 2. Суммы вероятностей вычисляются по следующим формулам:

$$\sum_{i\geqslant 2} p_i = p_1\frac{p^2}{q},$$

$$\sum_{i\geqslant 1} p_{i,i} = p_1\frac{q}{p(1-2pq)},$$

$$\sum_{i\geqslant 1}(i+1)p_{i,i} = p_1\frac{q(2-pq(3-\gamma))}{p(1-2pq)(1-pq(2-\gamma))},$$

$$\sum_{i\geqslant 1} ip_{i,i} = p_1\frac{q(1-pq)}{p(1-2pq)(1-pq(2-\gamma))},$$

$$\sum_{i\geqslant 2} p_{i,0} = p_1\frac{p^3}{q(q-p)}.$$

Заметим, что выражение для R не зависит от p_1. Само значение вероятности p_1 находится из последнего равенства системы (1) с использованием соотношения

$$p_1\left(\frac{(1-pq)^2}{p^2q} + \frac{q}{p(p^2+q^2)} + \frac{p^3}{q(q-p)}\right) = 1.$$

Приведённые формулы позволяют вычислить значение доли R при произвольных значениях параметров $0 \leqslant p < 1/2$ и $0 \leqslant \gamma \leqslant 1$. Результаты вычислений приведены на рис. 2–4.

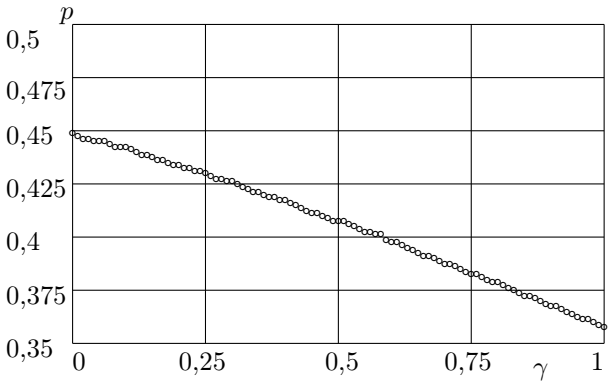

Рис. 2

На рис. 2 показан график зависимости от параметра γ минимального значения вероятности p, при котором впервые выполняется условие $R > 1/2$. Вычисления показывают, что выигрыш корыстной группы превышает при соответствующем значении γ выигрыш остальной группы при значениях вероятности p в пределах

$$0{,}358 \leqslant p \leqslant 0{,}449.$$

Наибольшее значение достигается при $\gamma = 0$, а наименьшее при $\gamma = 1$.

На рис. 3 показан аналогичный график зависимости от параметра γ минимального значения вероятности p, при котором впервые выполняется условие $R > p$. Получаем, что выигрыш корыстной группы при соответствующем значении γ превышает выигрыш, полученный ими при честном выполнении протокола, при значениях вероятности p в пределах

$$0 < p \leqslant 0{,}429.$$

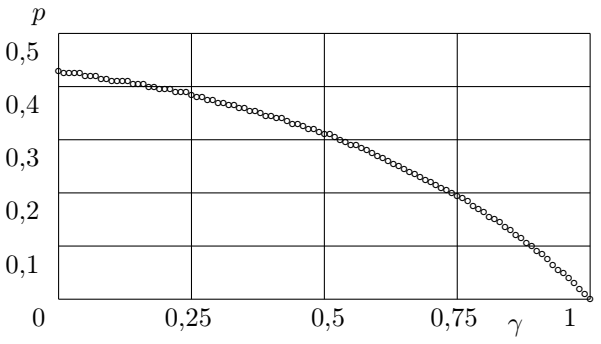

Рис. 3

В работе [1] этот интервал имеет вид $0 < p \leqslant 0{,}333$.

На графике рис. 4 приведена зависимость величины выигрыша R при честном и корыстном майнинге в зависимости от величины вероятности p для трёх значений параметра γ (0, 0,5 и 1).

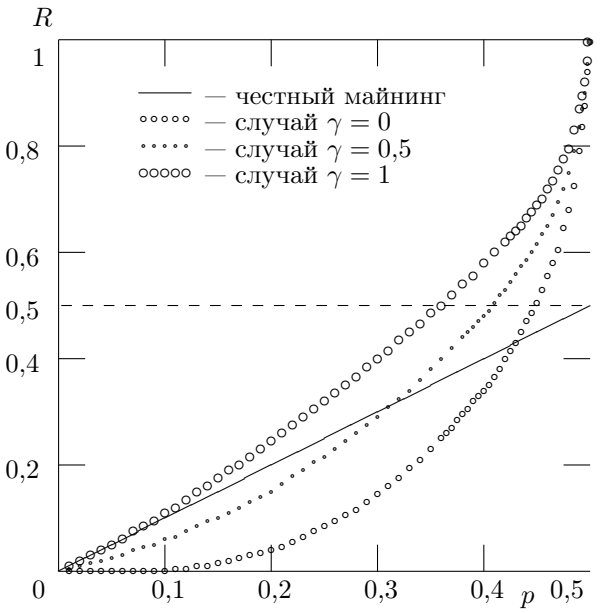

Рис. 4

Автор выражает благодарность рецензенту за внимательное прочтение рукописи и многочисленные полезные замечания.

ЛИТЕРАТУРА

1. *Ittay E. and Emin G. S.* Majority is Not Enough: Bitcoin Mining is Vulnerable. arXiv:1311.0243. 2013. `http://arxiv.org/abs/1311.0243`.
2. *Ittay E. and Emin G. S.* Majority is not enough: bitcoin mining is vulnerable // Financial Cryptography and Data Security: 18th Intern. Conf. Christ Church, Barbados, March 3–7, 2014. P. 436–454.
3. *Ittay E. and Emin G. S.* Majority is not enough: bitcoin mining is vulnerable // Commun. ACM. 2018. V. 61. No. 7. P. 95–102. `https://doi.org/10.1145/3212998`.

УДК 519.7 DOI 10.17223/2226308X/13/23

ON THE NUMBER OF UNSUITABLE BOOLEAN FUNCTIONS IN CONSTRUCTIONS OF FILTER AND COMBINING MODELS OF STREAM CIPHERS[1]

T. A. Bonich, M. A. Panferov, N. N. Tokareva

It is well known that every stream cipher is based on a good pseudorandom generator. For cryptographic purposes, we are interested in generation of pseudorandom sequences of the maximal possible period. A feedback register is one of the most known cryptographic primitives that is used in construction of stream generators. We analyze periodic properties of pseudorandom sequences produced by filter and combiner generators equipped with nonlinear Boolean functions. We determine which nonlinear functions in these schemes lead to pseudorandom sequences of not maximal possible period. We call such functions unsuitable and count the exact number of them for an arbitrary n.

Keywords: *stream cipher, filter generator, combiner generator, gamma, Boolean function.*

Remember that a *feedback shift register* (*FSR*) contains two parts: a binary block $x = (x_{n-1}, \dots x_0)$ of length n and a feedback function $f : (x_{n-1}, \dots, x_0) \to \{0, 1\}$, where f is a Boolean function in n variables. First, we fill the block x with concrete values of bits; together they form the *initial state* of the register. For functioning of the FSR, the time is considered to be discrete, i.e., it is divided into clock cycles. On each clock cycle, the value of $f(x)$ is calculated first, then the state $x = (x_{n-1}, \dots, x_1, x_0)$ of the register changes to the state $x' = (x_{n-2}, \dots, x_0, f(x))$, and the bit x_{n-1} is written as the first bit of the generated sequence *gamma*.

The properties of gamma generated by FSR are well studied in the case when f is a linear function. If f is nonlinear [1], then there are too many open questions with properties of gamma that all are connected to analysis of nonlinear recurrent sequences [2, 3]. That is why in cryptography some nonlinear *combinations* of linear FSRs are considered, for instance, filter and combining models of stream generators [4, 5].

In this paper, we analyze pseudorandom sequences produced by filter and combiner generators. Namely, we study which nonlinear functions h in these schemes lead to pseudorandom sequences such that their periods are not maximally possible. We call such functions *unsuitable* and count the exact number of them for an arbitrary n.

A *linear feedback shift register* (*LFSR*) consists of two parts: a binary vector $x = (x_{n-1}, \dots x_0)$ of length n and a linear feedback function f in n variables. A *state* of the register is a filling of vector x. During encryption, the register changes its states under an action of the feedback function. *Gamma* is a pseudorandom sequence generated by LFSR.

Also, LFSR can be specified using feedback polynomials. It is a polynomial of degree n defining bits to be summed. If $f(x_{n-1}, \dots, x_0) = a_0 x_{n-1} \oplus a_1 x_{n-2} \oplus \dots \oplus a_{n-1} x_0$, then the corresponding feedback polynomial is defined as $p(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_{n-1} z + 1$. If $p(z)$ is a primitive polynomial, then the period of a pseudorandom sequence generated by LFSR is maximal, i.e., is equal to $2^n - 1$. Thererfore, linear feedback shift registers are usually considered with primitive polinomials.

[1]The work is supported by Mathematical Center in Akademgorodok under agreement No. 075-15-2019-1613 with the Ministry of Science and Higher Education of the Russian Federation and Laboratory of Cryptography JetBrains Research.

1. Functions for the filter model

The filter generator consists of a single shift register of length n with a linear feedback and uses a primitive polynomial to change states. A Boolean function $h(x_{n-1}, \dots, x_0)$, applied to the current state, generates a pseudorandom sequence gamma.

Let $\gamma = (y_1 y_2 \dots y_{2^n-1})$, where $y_1 = h(x_{n-1}, \dots, x_0)$, $y_2 = h(x_{n-2}, \dots, x_0, f(x_{n-1}, \dots, x_0))$, etc. Since the number of all nonzero states is equal to $2^n - 1$, the maximal period of gamma is $2^n - 1$ too. In this paper, we would like do determine all Boolean function h in n variables that lead to gammas with non-maximum period. Let us call such functions *unsuitable*.

Note that the number of them does not depend on a linear feedback function. But whether the function is suitable or not for a given generator depends on the feedback function. When we count the number of unsuitable functions h, we do not consider a specific set of states. We say that there is a certain number of different states which the generator uses (all sets, that primitive polynomials generate, fit this definition). Next, we study which pseudorandom sequences have the maximum length. We analyze the number of unsuitable sequences and then the number of unsuitable functions. Thus, our reasonings do not affect the specific order of the states. Accordingly, for any set of states which the generator uses, there is the number of unsuitable functions h exactly that we calculated.

Theorem 1. Let n be an integer and $2^n - 1 = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_s^{\alpha_s}$, where p_i are distinct prime numbers, α_i are positive integers, s is a some number. Then the number of unsuitable Boolean functions in n variables for the filter generator with LFSR based on a primitive polynomial is equal to

$$2 \sum_{\beta \in \mathbb{F}_2^s, \beta \neq 0} (-1)^{\beta_1 + \dots + \beta_s + 1} 2^{p_1^{\alpha_1 - \beta_1} \dots p_s^{\alpha_s - \beta_s}},$$

where $\beta = (\beta_1, \dots, \beta_s)$.

2. Functions for the combining model

Combiner generators use several linear feedback shift registers. Each register has its own length n_i and uses its primitive polynomial for changing states. A Boolean function $h(X_1, \dots, X_m)$ generates the pseudorandom sequence gamma where X_i is a register bit string i. Since we do not use the zero state in combiner generator, the total number of states does not exceed $(2^{n_1} - 1)(2^{n_2} - 1) \dots (2^{n_m} - 1)$. In this case, the maximum is reached at $\gcd(n_i, n_j) = 1$, where $i, j = 1, \dots, m$, $i \neq j$, and if all LFSRs have primitive feedback polynomials. Then the Boolean function can generate a gamma with period from 1 to $(2^{n_1} - 1)(2^{n_2} - 1) \dots (2^{n_m} - 1)$.

We consider a more general model of a combiner generator that is applied in ciphers Grain [6] and Bean [7]. Note that the classical combining model does not allow to describe a number of modern stream ciphers based on the more complicated operating with bits from different registers. In this case, the combiner generator, in which the function depends only on the extreme bits of the registers, is included in the model we consider. In a nonlinear model sometimes it is more convenient to work with several smaller registers than with one large. It should be noted that the model that we consider can be used not only in cases of all linear or all non-linear registers, but also in cases of mixed registers (i.e., some registers are linear, some are non-linear).

Theorem 2. Let n be an integer, $\sum_{i=1}^{m} n_i = n$, $(2^{n_1} - 1)(2^{n_2} - 1) \dots (2^{n_m} - 1) = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_s^{\alpha_s}$, where p_i are different prime numbers, $\alpha_i > 0$, s is an integer. Then the

number of unsuitable Boolean functions in n variables for the combiner generator with LFSRs of lengths $n_1, \ldots, n_m$ all based on primitive polynomials is equal to

$$2^{2^{n_1+n_2+\cdots+n_m}-(2^{n_1}-1)(2^{n_2}-1)\ldots(2^{n_m}-1)} \sum_{\beta\in\mathbb{F}_2^s,\beta\neq 0} (-1)^{\beta_1+\cdots+\beta_s+1} 2^{p_1^{\alpha_1-\beta_1}\ldots p_s^{\alpha_s-\beta_s}},$$

where $\beta = (\beta_1, \ldots, \beta_s)$.

3. Functions for models with nonlinear registers

A *nonlinear feedback shift register* (*NFSR*) consists of two parts: a binary vector $x = (x_{n-1}, \ldots x_0)$ of length n and a nonlinear state function $f : (x_{n-1}, \ldots, x_0) \to \{0, 1\}$ in n variables.

Similarly to the linear case, consider the filter generator. We assume that NFSR passes over all 2^n states, i.e., it has maximal possible period.

Theorem 3. Let n be an integer. Then the number of unsuitable Boolean functions in n variables for the filter generator with NFSR of the maximal possible period is equal to $2^{2^{n-1}}$.

There is an another question related to NFSRs: how to determine for which nonlinear feedback functions NFSR of length n has the maximal possible period 2^n? This question is hard and still open.

We kindly thank the reviewer for careful reading of our paper and significant remarks.

REFERENCES

1. *Key E.* An analysis of the structure and complexity of nonlinear binary sequence generators. IEEE Trans. Inform Theory, 1976, no. 22, pp. 732–736.
2. *Gluhov M. M., Elizarov V. P., Nechaev A. A.* Algebra [Algebra]. Moscow, Gelios ARV Publ., 2003. (in Russian)
3. *Roman'kov V. A.* Vvedenie v kriptografiyu [Introduction to Cryptography]. Moscow, Forum Publ., 2012. (in Russian)
4. *Tokareva N. N.* Simmetrichnaya kriptografiya. Kratkiy kurs [Symmetric Cryptography. A Short Course]. Novosibirsk, NSU Publ., 2012.
5. *Carlet C.* Boolean functions for cryptography and error-correcting codes. Eds. P. Hammer and Y. Crama. Boolean Models and Methods in Mathematics, Computer Science, and Engineering. Cambridge, Cambridge Univ. Press, 2010. Ch. 8, pp. 257–397. `www.math.univ-paris13.fr/~carlet/`.
6. *Hell M., Johansson T., and Meier W.* A stream cipher for constrained environments. Int. J. Wireless Mobile Comput., 2007, vol. 2, no. 1, pp. 86–93.
7. *Kumar N., Ojha S., Jain K., and Lal S.* BEAN: A lightweight stream cipher. Proc. 2nd Intern. Conf. SIN'2009, ACM, 2009, pp. 168–171.

UDC 621.391.7 DOI 10.17223/2226308X/13/24

EFFICIENT S-REPETITION METHOD FOR CONSTRUCTING AN IND-CCA2 SECURE MCELIECE MODIFICATION IN THE STANDARD MODEL

Y. V. Kosolapov, O. Y. Turchenko

The paper is devoted to the construction of IND-CCA2-secure modification of the McEliece cryptosystem in the standard model. The modification uses S-repetition

encryption of $S/2$ various messages with one common secret permutation, in contrast to other modifications that use S-repetition encryption of one message. Thus, this modification provides IND-CCA2-security with an efficient information transfer rate.

Ключевые слова: *post-quantum cryptography, McEliece-type cryptosystem,* IND-CCA2-*security, S-repetition encryption.*

1. Introduction

Currently, much effort is being devoted to the development of quantum computers. Therefore, the study of post-quantum cryptosystems is an important task. One suitable scheme in the post-quantum era is the McEliece cryptosystem [1]. Note that the McEliece cryptosystem does not use quantum mechanical properties. However, the original McEliece scheme is vulnerable to attacks on cyphertexts. To date, many approaches have been developed to modify the McEliece cryptosystem. One of the most successful approaches is based on the application of correlated products [2]. For instance, in [3, 4] authors presented IND-CCA2-secure modifications in the standard model. At the same time, the main idea of correlated products is not effective in practice, because it requires to transmit S encrypted blocks for one information message. Based on the ideas from [3], we offer a new IND-CCA2-secure modification of the McEliece cryptosystem in the standard model, which requires to transmit S encrypted blocks for $S/2$ information messages.

2. Preliminaries

Let n, t be natural, $2t < n$, $[n] = \{1, \ldots, n\}$, $\beta \subseteq [n]$, $2^{[n]}$ is set of all subsets of $[n]$, $\mathbb{F}_2$ be a Galois field of cardinality 2. The support of the vector $\mathbf{m} = (m_1, \ldots, m_n) \in \mathbb{F}_2^n$ is the set $\mathrm{supp}(\mathbf{m}) = \{i : m_i \neq 0\}$ and the Hamming weight of this vector is a number $\mathrm{wt}(\mathbf{m}) = |\mathrm{supp}(\mathbf{m})|$. A function $\gamma : \mathbb{N} \to [0, 1]$ is negligible of k, if

$$\forall c \in \mathbb{N}\ \exists k_c \in \mathbb{N}\ \forall k > k_c\, (\gamma(k) \leqslant k^{-c}).$$

We will use the notations similarly to the [3]. If S is a finite set, then $s \in_R S$ denotes the operation of picking an element at random and uniformly from S. Denote by $\mathcal{E}_{n,t,\beta}$ the subset of $\mathbb{F}_2^n$ such that any vector $\mathbf{e} = (e_1, \ldots, e_n) \in \mathcal{E}_{n,t,\beta}$ has Hamming weight t and $e_i = 0$ for any $i \in \beta$. We will write $\mathcal{E}_{n,t}$ when $\beta = \varnothing$. Let us define a cryptosystem as triplet of algorithms, i.e. $\Sigma = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where:

1) $\mathcal{K}$ is a probabilistic polynomial-time key generation algorithm which takes as input a security parameter $N \in \mathbb{N}$ and outputs a pair of public-key and a secret-key (pk, sk);
2) $\mathcal{E}$ is probabilistic polynomial-time encryption algorithm which takes as input a public-key pk and a message $\mathbf{m}$ and outputs a ciphertext $\mathbf{c}$; we will write $\{\mathbf{m}\}_{pk}^{\Sigma}$ as encryption of the message $\mathbf{m}$ with the key pk;
3) $\mathcal{D}$ is deterministic polynomial-time decryption algorithm which takes as input a secret-key sk and a ciphertext $\mathbf{c}$ and outputs either a message $\mathbf{m}$ or a symbol $\perp$ in the case, when the ciphertext is incorrect; decryption of the ciphertext $\mathbf{c}$ on the secret key sk we will denote $\{\mathbf{c}\}_{sk}^{\Sigma}$.

Let us define signature scheme (SS) and one-time strongly unforgeable feature in the same way as [3]. A signature scheme is triplet of algorithms $\mathrm{SS} = (\mathcal{K}_{\mathrm{SS}}, Sign, Check)$, where $\mathcal{K}$ is key generation algorithm which takes as input a security parameter $N \in \mathbb{N}$ and outputs a signing-key $\mathbf{dsk}$ and a verification-key $\mathbf{vk}$, $Sign$ is signing algorithm which takes as input a signing-key $\mathbf{dsk}$ and a message $\mathbf{m}$ and outputs a signature σ, $Check$ is checking algorithm which takes as input a verification-key $\mathbf{vk}$ a message $\mathbf{m}$ and a signature σ and outputs 1 if

σ is valid for $\mathbf{m}$ and 0 otherwise. It is important to note, that one-time strongly unforgeable signature scheme can be constructed using one-way functions (see [5, 6]).

Consider the McEliece cryptosystem as a triplet of polynomial-time algorithms: $\mathrm{McE} = (\mathcal{K}_{\mathrm{McE}}, \mathcal{E}_{\mathrm{McE}}, \mathcal{D}_{\mathrm{McE}})$ on the linear $[n, k, d]$-code $C \subseteq \mathbb{F}_2^n$, where n is the length, k is the code dimension, and d is the minimum code distance. Let G be the generator matrix of the code C, $t = \lfloor (d-1)/2 \rfloor$. A secret key sk is a pair (S, P), where S is a non-singular $(k \times k)$-matrix over the field $\mathbb{F}_2$ and P is a permutation $(n \times n)$-matrix. A public key pk is a pair $(\widetilde{G} = SGP, t)$. Encryption of a message $\mathbf{m} \in \mathbb{F}_2^k$ is performed according to the rule

$$\{\mathbf{m}\}_{pk}^{\mathrm{McE}} = \mathbf{m}\widetilde{G} + \mathbf{e} = \mathbf{c},\ \mathbf{e} \in_R \mathcal{E}_{n,t}.$$

To decrypt the ciphertext $\mathbf{c}$, one should use an effective decoder $\mathrm{Dec}_C : \mathbb{F}_2^n \to \mathbb{F}_2^k$ of the code C and the secret key sk:

$$\{\mathbf{c}\}_{sk}^{\mathrm{McE}} = \mathrm{Dec}_C(\mathbf{c}P^{-1})S^{-1}.$$

3. Efficient S-repetition construction

On the basis of the Randomized McEliece cryptosystem [7] we construct a new cryptosystem $\mathrm{bMcE}_l = (\mathcal{K}_{\mathrm{bMcE}_l}, \mathcal{E}_{\mathrm{bMcE}_l}, \mathcal{D}_{\mathrm{bMcE}_l})$ and call it the basic cryptosystem. For the vector $\mathbf{m}(\in \mathbb{F}_q^k)$ and the ordered set $\omega = \{\omega_1, \ldots, \omega_l\} \subseteq [k]$, where $\omega_1 < \ldots < \omega_l$, we consider the projection operator $\Pi_\omega : \mathbb{F}_q^k \to \mathbb{F}_q^{|\omega|}$ acting according to the rule: $\Pi_\omega(\mathbf{m}) = (m_{\omega_1}, \ldots, m_{i_{\omega_l}})$. For ω consider a subset $\mathcal{G}(\omega)$ of permutations group $\mathcal{S}_k$ acting on the elements of the set $[k]$:

$$\mathcal{G}(\omega) = \{\pi \in \mathcal{S}_k : \pi(1) = \omega_1, \ldots, \pi(l) = \omega_l\}.$$

With every permutation π from $\mathcal{G}(\omega)$ we associate a permutation $(k \times k)$-matrix R_π. The encryption rule of basic McEliece bMcE_l has the form

$$\{\mathbf{m}\}_{pk,\omega}^{\mathrm{bMcE}_l} = \{(\mathbf{m} \parallel \mathbf{r}_1)R_\pi\}_{pk}^{\mathrm{McE}} \parallel \{(\mathbf{m} \parallel \mathbf{r}_2)R_\pi\}_{pk}^{\mathrm{McE}} = \mathbf{c}_1 \parallel \mathbf{c}_2 = \mathbf{c},$$

where $\mathbf{m} \in \mathbb{F}_q^l$, $\omega \subset_R [k]$, $|\omega| = l$, $\mathbf{r}_1 \in_R \mathbb{F}_q^{k-l}$, $\mathbf{r}_2$ is formed in accordance with the restriction $\mathrm{supp}(\mathbf{r}_1 - \mathbf{r}_2) = [k] \backslash \omega$, $\pi \in_R \mathcal{G}(\omega)$. The error vectors $\mathbf{e}_1$ and $\mathbf{e}_2$, generated in McE-encryption, are chosen such that $\mathbf{e}_1 \in_R \mathcal{E}_{n,t}$, $\mathbf{e}_2 \in_R \mathcal{E}_{n,t,\mathrm{supp}(\mathbf{e}_1)}$. From here, it follows that

$$\mathrm{wt}(\mathbf{e}_1) + \mathrm{wt}(\mathbf{e}_2) = 2t.$$

To decrypt the ciphertext $\mathbf{c}$, one should calculate

$$\{\mathbf{c}\}_{sk}^{\mathrm{bMcE}_l} = \Pi_\eta(\{\mathbf{c}_1\}_{sk}^{\mathrm{McE}}),\ \eta = [k] \setminus \mathrm{supp}(\{\mathbf{c}_1\}_{sk}^{\mathrm{McE}} - \{\mathbf{c}_2\}_{sk}^{\mathrm{McE}}). \tag{1}$$

Using the one-time strongly unforgeable signature scheme $\mathrm{SS} = (\mathcal{K}_{\mathrm{SS}}, Sign, Check)$ we will construct a new S-repetition McEliece cryptosystem as a triplet of polynomial-time algorithms: $\mathrm{bMcE}_l^s = (\mathcal{K}_{\mathrm{bMcE}_l^s}, \mathcal{E}_{\mathrm{bMcE}_l^s}, \mathcal{D}_{\mathrm{bMcE}_l^s})$. Key generation algorithm $\mathcal{K}_{\mathrm{bMcE}_l^s}$ takes as input a security parameter $N \in \mathbb{N}$ and outputs a public-key pk and a secret key sk of the form

$$pk = ((pk_i^0, pk_i^1))_{i=1}^s,\ sk = ((sk_i^0, sk_i^1))_{i=1}^s,$$

where $pk_i^b, sk_i^b \leftarrow \mathcal{K}_{\mathrm{McE}}(N)$, $b \in \{0, 1\}$, $i \in [s]$.

To define encryption algorithm, let us consider a message $\mathbf{m} = (\mathbf{m}_1 \parallel \ldots \parallel \mathbf{m}_s)$ where $\mathbf{m}_i \in \mathbb{F}_2^l$. Encryption algorithm $\mathcal{E}_{\mathrm{bMcE}_l^s}$ takes as input a public-key pk and a message $\mathbf{m}$ and outputs a ciphertext $\mathbf{c}$:

$$\mathbf{c} = \{\mathbf{m}\}_{pk^{\mathbf{vk}}}^{\mathrm{bMcE}_l^s} = \mathbf{c}' \parallel \mathbf{vk} \parallel \sigma,$$

where $(\mathbf{dsk}, \mathbf{vk}) \leftarrow \mathcal{K}_{\mathrm{SS}}(N)$, $\mathbf{vk} = (vk_1, \ldots, vk_s)$, $\sigma = Sign(\mathbf{dsk}, \mathbf{c}')$, $pk^{\mathbf{vk}} = (pk_1^{vk_1}, \ldots, pk_s^{vk_s})$, and $\mathbf{c}'$ calculated as follows:

$$\mathbf{c}' = \mathbf{c}'_1 \parallel \ldots \parallel \mathbf{c}'_s = [\mathbf{c}'_{1,1} \parallel \mathbf{c}'_{1,2}] \parallel \ldots \parallel [\mathbf{c}'_{s,1} \parallel \mathbf{c}'_{s,2}],$$

where $\mathbf{c}'_j = [\mathbf{c}'_{j,1} \parallel \mathbf{c}'_{j,2}] = \{\mathbf{m}_j\}_{pk_j^{\mathbf{vk}_j},\omega}^{\mathrm{bMcE}_l}$ for $j \in [s]$ and ω is chosen randomly once for all $j = 1, \ldots, s$.

Decryption algorithm $\mathcal{D}_{\mathrm{bMcE}_l^s}$ takes as input a secret-key sk and a ciphertext $\mathbf{c}$ and outputs either a message $\mathbf{m} \in \mathbb{F}_q^{sl}$ or a error symbol $\perp$. On the first step, $\mathcal{D}_{\mathrm{bMcE}_l^s}$ checks signature of the message. If $Check(\mathbf{c}', \mathbf{vk}, \sigma) = 0$, then $\mathcal{D}_{\mathrm{bMcE}_l^s}$ outputs $\perp$, otherwise it computes $\mathbf{m}$ as follows. For each $\mathbf{c}'_i$ from $\mathbf{c}' = \mathbf{c}'_1 \parallel \ldots \parallel \mathbf{c}'_s$ it finds $\mathbf{m}_i = \{\mathbf{c}'_i\}_{sk_i}^{\mathrm{bMcE}_l}$ and η_i according to (1) and outputs

$$\mathbf{m} = \begin{cases} \mathbf{m}_1 \parallel \ldots \parallel \mathbf{m}_s, & \text{if } \eta_1 = \ldots = \eta_s, \\ \perp, & \text{otherwise.} \end{cases}$$

Let McE be the McEliece cryptosystem with security parameter N. The security of McE is based on two following standard assumptions.

Assumption 1. There is no polynomial algorithm capable of distinguishing the $(k \times n)$-matrix of the public key of the McE cryptosystem from a random $(k \times n)$-matrix with non-negligible probability in N.

Assumption 2. There is no polynomial algorithm that solves the problem of decoding a general linear code.

According to [8], the problem of decoding a general linear code is NP-hard. Since $P \neq NP$ has not been proved, we formulate this only as an assumption.

Note that, if these assumptions hold, then one can say that McE is one way trapdoor function (or OW-CPA secure) [9]. The hardness of most McE-type cryptosystems is based on the above assumptions (for example, [3, 4, 7]). To formulate the following theorem we should introduce auxiliary assumption.

Assumption 3. There is no polynomial algorithm that takes as input ciphertext $\mathbf{c}$ of the McE and the number $l \in \mathbb{N}$, and outputs 0 if $\mathbf{c}$ corresponds to an information message of a weight less than l and outputs 1 if $\mathbf{c}$ corresponds to an information message of weight l with non-negligible distinguishing advantage in the N.

Theorem 1. Let SS be one-time strongly unforgeable signature scheme. Then bMcE_l^s with security parameter N and fixed s is IND-CCA2 secure if assumptions 1–3 hold.

REFERENCES

1. *McEliece R. J.* A public-key cryptosystem based on algebraic coding theory // DSN Progress Report. 1978. P. 42–44.
2. *Rosen A. and Segev G.* Chosen-ciphertext security via correlated products // LNCS. 2009. V. 5444. P. 419–436.

3. *Dottling N., Dowsley R., Quade J. M., and Nascimento A. C. A.* A CCA2 secure variant of the McEliece cryptosystem // IEEE Trans. Inform. Theory. 2012. V. 58(10). P. 6672–6680.
4. *Persichetti E.* On a CCA2-secure variant of McEliece in the standard model // Provable Security. 2018. V. 11192. P. 165–181.
5. *Lamport L.* Constructing Digital Signatures from One-Way Functions. SRI International, 1979. `https://www.microsoft.com/en-us/research/publication/constructing-digital-signatures-one-way-function/`
6. *Naor M. and Yung M.* Universal One-Way Hash Functions and their Cryptographic Applications // Proc. STOC'89. N.Y.: ACM, 1989. P. 33–43.
7. *Nojima R., Imai H., Kobara K., et al.* Semantic security for the McEliece cryptosystem without random oracles // Designs, Codes and Cryptography. 2008. V. 49. P. 289–305.
8. *Berlekamp E. R., McEliece R. J., and van Tilborg H. C.* On the inherent intractability of certain coding problems // IEEE Trans. Inform. Theory. 1978. V. 24. No. 3. P. 384–386.
9. *Kobara K. and Imai H.* On the one-wayness against chosen-plaintext attacks of the Loidreau's modified McEliece PKC // IEEE Trans. Inform. Theory. 2003. V. 49. No. 12. P. 3160–3168.

Секция 4

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

УДК 004.056.53, 004.032.26 DOI 10.17223/2226308X/13/25

НЕЙРОСЕТЕВАЯ ОБФУСКАЦИЯ ВЫЧИСЛЕНИЙ НАД ЗАШИФРОВАННЫМИ ДАННЫМИ

В. Л. Елисеев

Предложен подход по нейросетевой криптографической обфускации вычислений. Опираясь на ранее полученные результаты о свойстве строгой обфускации неразличимости для нейросетевого аппроксиматора, мы предлагаем использовать нейросети для выполнения арифметических и других операций над зашифрованными данными, реализуя таким образом идею применения гомоморфного шифрования для выполнения доверенных вычислений в недоверенной среде. Проводится оценка криптографических свойств предложенного механизма и сопоставление с традиционными подходами к шифрованию на основе секретного ключа. Обсуждаются достоинства и недостатки нейронных сетей применительно к задачам обфускации и обработки зашифрованных данных.

Ключевые слова: *искусственная нейронная сеть, обфускация, гомоморфное шифрование, оценка стойкости.*

Введение

Тенденцией настоящего времени в криптологии является активное исследование новых криптографических систем, обладающих новыми свойствами. Множество работ посвящено изучению так называемых постквантовых алгоритмов, обеспечивающих стойкость к современным и перспективным угрозам. Большое внимание уделяется также функциональному и гомоморфному шифрованию, основной целью которых является выполнение некоторых типов операций над зашифрованными данными без расшифрования. Для обеспечения требуемых свойств задействованы многие математические формализмы, прежде не использовавшиеся в криптографии. Рассмотрим механизм искусственных нейронных сетей для реализации цели, преследуемой гомоморфной криптографией, — выполнению вычислений над зашифрованными данными.

1. Обзор

1.1. Искусственные нейронные сети

Исследования биологических нейронных сетей в 30-е годы XX века дали многие из идей, лёгших в основу кибернетики. Наиболее прямолинейной была попытка получить свойства биологических нейронных сетей с помощью формальных математических моделей, получивших обобщённое название «*искусственные нейронные сети*». Одной из успешных формальных моделей биологических нейронных сетей стал *многослойный перцептрон* (*MultiLayer Perceptron* (*MLP*)). Многослойный перцептрон (далее — нейросеть) вычисляет выходной вектор $x^{(N)}$ по входному $x^{(0)}$:

$$x^{(N)} = \mathcal{N}(x^{(0)}),$$

при этом реализует алгебраическое преобразование вида

$$x_i^{(k)} = s\left(\sum_{j=1}^{m_{k-1}} w_{ij}^{(k)} x_j^{(k-1)}\right),\ k = 1, \ldots, N,\ i = 1, \ldots, m_k,$$

где N — количество слоёв нейросети; $x^{(k)} \in \mathbb{R}^{m_k}$ — вектор выходов k-го слоя; $x^{(0)} \in \in \mathbb{R}^{m_0}$ — вектор входов нейросети; m_k — количество нейронов слоя k; m_0 — количество входов первого слоя нейросети; $w_{ij}^{(k)} \in \mathbb{R}$ — весовой коэффициент j-го входа i-го нейрона слоя k; $s(.)$ — дифференцируемая функция, называемая также функцией активации. В зависимости от вида функции активации различают линейный ($s(t) = t$), ступенчатый ($s(t) = \text{sign}(t)$), сигмоидальный $\left(s(t) = \dfrac{1}{1+e^{-t}}\right)$ нейроны. В глубоких нейросетях используются и другие виды функций активации [1].

Для многослойного персептрона с двумя слоями и сигмоидальной функцией активации в 1987 г. Р. Хехт-Нильсеном доказана представимость любой непрерывной функции $f(\cdot)$ с любой наперёд заданной точностью [2]. Таким образом, можно утверждать, что многослойный перспептрон с сигмоидальной функцией активации подходит для реализации любых сколь угодно сложных функций, в том числе булевых, широко используемых в криптографии.

Для аппроксимации некоторой целевой функции $f(\cdot)$ нейросетью $\mathcal{N}(\cdot)$ применяется процедура, называемая обучением, результатом которой являются такие значения весовых коэффициентов нейронной сети $w_{ij}^{(k)}$, что ошибка аппроксимации $\varepsilon = \| \mathcal{N}(x) - f(x) \|$ оказывается в допустимых пределах. Процедура обучения, как правило, реализуется путём многомерной оптимизации заданной функции ошибки в пространстве весовых коэффициентов на некотором множестве обучающих данных, образованных парами $(x, f(x))$. Оптимизируемой функцией часто выступает среднеквадратическая ошибка, вычисляемая по формуле

$$\overline{\varepsilon^2} = \sqrt{\frac{1}{M} \sum_{i=1}^{M} \| \mathcal{N}(x_i) - f(x_i) \|^2}.$$

Следует отметить, что в большинстве случаев нейросети используются для аппроксимации *неизвестной* функции $f(\cdot)$, про которую известно только то, что она для некоторых аргументов $x \in \mathbb{X}$ принимает значения $y \in \mathbb{Y}$, однако вид функции и её поведение (за исключением непрерывности) между точками (x, y) неизвестны. Аппроксимируя неизвестную функцию по таблично заданному подмножеству её значений, нейросеть позволяет решать многие задачи, связанные с классификацией, распознаванием образов, фильтрацией помех, предсказанием временных рядов и даже моделированием квантовых процессов.

При обучении нейросети на ограниченном множестве данных и при неизвестном виде аппроксимируемой функции приходится решать ряд проблем, связанных с выбором архитектуры нейросети (число слоёв и число нейронов в них) и эффектом переобучения (overfitting) — потерей обобщающей способности нейросетью [3]. Для борьбы с переобучением исходное множество пар (x, y) разделяют на обучающие и проверочные, причём вторые не участвуют в обучении и используются для получения оценки ошибки ε.

Сигмоидальная функция активации $s(t) = (1 + e^{-t})^{-1}$ непрерывна и ограничена областью значений $0 < s(t) < 1$, поэтому выходы нейросети не могут быть интерпретированы как булевы. Поскольку наша задача — представление векторных булевых

функций, то будем приводить выход нейросети к булеву виду с помощью функции $b : \mathbb{R} \to \mathbb{B}$, где $\mathbb{B} = \{0, 1\}$:

$$b(x) = \begin{cases} 0, & x < 0{,}5, \\ 1, & x \geqslant 0{,}5. \end{cases}$$

Таким образом, нейросетевая обфускация векторной булевой функции $f(\cdot)$ должна реализовываться последовательным вычислением нейросетью вектора непрерывных значений $x^{(N)}$ и его преобразованием к вектору булевых значений:

$$x^{(N)} = \mathcal{N}(x^{(0)}), \quad y_i = b(x_i^{(N)}),\ i = 1, \ldots, m_N.$$

1.2. Нейронные сети и криптология

Проведём краткий обзор известных примеров применения искусственных нейросетей в задачах разработки и анализа криптографических примитивов. Известны работы, посвящённые криптоанализу с использованием нейросетей [4, 5], однако нейросеть в этом случае выступает в качестве вспомогательного инструмента и не используется для решения задачи защиты информации. В некоторых работах рассматривается создание на основе нейросетей хэш-функций [6, 7] и генераторов случайных чисел [8].

Делались также попытки построить криптосистемы на основе специфических способностей нейронных сетей к обучению и представлению нелинейных зависимостей, однако такие криптосистемы оказались довольно быстро взломаны [9, 10]. Перечень нейросетевых криптосистем и их взломов приведён в табл. 4 в [11].

В работе [12] решается задача синтеза нейросетевой симметричной криптосистемы по критерию минимизации вероятности дешифрования подслушивателем передаваемого сообщения при фиксированном ключе. В качестве подслушивателя рассматривается ещё одна нейросеть, поэтому предложенный подход получил название Adversarial Neural Cryptography (ANC). Отличие нейросети подслушивателя от нейросетей легитимных контрагентов заключается в отсутствии входов с битами секретного ключа. Предложенный алгоритм обучения позволил синтезировать нейросети для шифрования и расшифрования данных. Однако успешное решение поставленной задачи не обеспечило секретность шифруемых данных в критериях, привычных для традиционной криптографии, поскольку роль подслушивателя выполняет нейросеть, что является очень упрощённой моделью криптоаналитика.

С целью устранения отмеченного недостатка в [11] предложено усилить секретность синтезируемого нейросетевого алгоритма шифрования/расшифрования за счёт возможности подслушивателя на этапе обучения использовать атаку с подобранным открытым текстом (Chosen Plaintext Attack). Вторым усовершенствованием было использование нейросети специальной архитектуры, названной *CryptoNet* и обеспечивающей операцию XOR на уровне базового элемента сети, что теоретически позволяет синтезировать в результате обучения даже одноразовый блокнот (one-time pad). Полученные результаты подтвердили возможность синтеза более секретной системы шифрования на основе нейросетей, однако практическая ценность пока невелика, так как в силу большой вычислительной сложности алгоритма обучения нейросети размер блока шифруемых данных и длина используемого ключа не превышали 16 бит.

Более масштабными и практически полезными выглядят попытки применить нейросети для обработки зашифрованных данных без ограничений на вид операций над ними. В основополагающей работе [13] доказывается возможность использования нейросетей для обработки зашифрованных данных через представление реализуемой нейросетью функции полиномом и дальнейшую реализацию этого полинома с помощью

гомоморфного шифрования (HE). Следует отметить некоторую избыточность приведённых авторами [13] рассуждений, поскольку доказываемый ими факт представимости любых непрерывных функций с помощью нейросетей типа «многослойный перцептрон» давно известен [14]. Тем не менее в [13] проводится важная мысль о подобии возможностей, предоставляемых гомоморфным шифрованием и искусственными нейросетями.

Недавние успехи в развитии и всё более эффективной реализации алгоритмов полного гомоморфного шифрования (FHE) сделали актуальной задачу обработки сложных данных в зашифрованной форме. Одним из инструментов обработки медицинских и финансовых данных являются нейросети, однако такие данные в большинстве случаев не должны раскрываться для третьей стороны — владельца вычислительных ресурсов. По этой причине авторы [15] предлагают компилятор, преобразующий обученную нейросеть в эквивалентную, но работающую над зашифрованными данными. Таким образом, довольно прямолинейный подход авторов [13] по замене нейросетей их гомоморфными полиномиальными аппроксимациями предложен в [15] для практической реализации.

Вопрос о криптостойкости использования нейросетей при операциях над зашифрованными данными в [13, 15] не рассматривается, видимо, из-за надежды на криптостойкость применяемого гомоморфного шифрования. Тем не менее можно утверждать, что криптостойкость в наилучшем случае будет определяться разрядностью используемого ключа гомоморфной криптосистемы.

Резюмируя, можно отметить, что наиболее привлекательными свойствами нейросетей, используемыми в исследованиях, являются обучение на примерах, хаотическая динамика и нелинейное отображение. При этом многие исследователи явно или неявно подразумевают, что реализованные в обученной нейросети преобразования не могут быть извлечены для полезного использования [12].

В [16] доказано, что нейросети обладают свойством обфускатора неразличимости, что позволяет использовать их для сокрытия деталей реализации вычислительных алгоритмов. Основная идея, предлагаемая в настоящей работе, заключается в применении нейронных сетей в качестве стойкого обфускатора алгоритмов и основанного на этом подхода по обработке зашифрованных данных непосредственно нейросетью.

2. Нейросетевая криптографическая обфускация

В качестве удобной абстракции некоторого алгоритма будем рассматривать программу P, реализующую вычисления над булевыми векторами фиксированной длины:

$$\forall x \in X \subseteq \mathbb{B}^n \ (y = P(x),\ y \in Y \subseteq \mathbb{B}^m).$$

Предположим, что программу P необходимо выполнять в недоверенной вычислительной среде, то есть существует риск того, что входные и выходные данные программы, а также, возможно, алгоритм и промежуточные результаты станут доступны злоумышленнику. К подобного рода вычислительным средам можно отнести все без исключения публичные облачные сервисы, доверие к которым основывается исключительно на репутации владельцев, а также технических мерах по разграничению доступа к данным различных пользователей сервиса.

Рассмотрим возможность обработки зашифрованных данных программой P с получением результата также в зашифрованном виде. Для этого определим операции зашифрования E и расшифрования D. Данные функции должны обеспечивать обратимость на полном множестве булевых векторов требуемой длины как для входа, так

и для выхода программы:

$$\forall x \in X \subseteq \mathbb{B}^n \; \exists c_x \in C_x \subseteq \mathbb{B}^n \; (c_x = E_X(x) \;\&\; x = D_X(c_x)),$$
$$\forall y \in Y \subseteq \mathbb{B}^m \; \exists c_y \in C_y \subseteq \mathbb{B}^m \; (c_y = E_Y(y) \;\&\; y = D_Y(c_y)).$$

В таком случае вычисления над зашифрованными данными могли бы быть представлены как

$$c_y = E_Y\Big(P\big(D_X(c_x)\big)\Big),$$

где $_x$ — зашифрованный входной вектор x; c_y — зашифрованный результат y. Операции зашифрования $c_x = E_X(x)$ и расшифрования $y = D_Y(c_y)$ должны выполняться в доверенном окружении.

Эта схема могла бы быть очевидным решением исходной задачи для выполнения программы в недоверенной среде, если бы не два обстоятельства. Во-первых, операции E_Y и D_X являются симметричными криптографическими алгоритмами, что неизбежно потребует хранения рядом с ними секретных ключей, что невозможно сделать доверенным образом в недоверенной среде. Во-вторых, программа P всё равно обрабатывает открытые данные, что предоставляет злоумышленнику достаточно возможностей для её исследования и, например, нелицензионного использования.

Согласно теореме 1 в [16], для любой векторной булевой функции возможно построить нейросетевой обфускатор $\mathcal{N}$, в частности, такой, что

$$\forall x \in X \; \big(c_x = E_X(x) \;\&\; c_y = E_Y(P(x)) \;\&\; \hat{c}_y = \mathcal{N}(c_x)\big) \Rightarrow \big(|c_{yi} - \hat{c}_{yi}| < 0{,}5, \; i = 1, \ldots, m\big).$$

Нейросетевой обфускатор реализует операцию вычисления зашифрованного результата по зашифрованным входным данным. При этом, согласно теореме 2 в [16], он является обфускатором неразличимости, то есть эффективно скрывает способ реализации целевой функции. Таким образом, обученная нейронная сеть $\mathcal{N}$ может выполняться в любом недоверенном окружении, выполняя над передаваемыми ей зашифрованными входными данными операцию, результат которой после расшифрования эквивалентен вычислению исходной программы P над открытыми входными данными.

Зададим последовательность шагов для обучения нейросетевого обфускатора программы, обрабатывающего зашифрованные данные (алгоритм 1).

Для выполнения вычислений над $x \in X$ в недоверенном окружении необходимо выполнить последовательность шагов, представленную алгоритмом 2. При необходимости многих вычислений шаг 1 можно выполнить однократно, повторяя только шаги 2–6.

Алгоритм 1. Обучение нейросетевого криптографического обфускатора

1: Вычислить результат работы программы P для всего множества возможных аргументов $x^{(j)} \in X$, образовав обучающее множество пар $(x^{(j)}, y^{(j)})$: $y^{(j)} = P(x^{(j)})$, $j = 1, \ldots, |X|$.
2: Создать взаимно однозначную таблицу подстановки $E_X : x^{(j)} \to c_x^{(j)}$, где $c_x^{(j)} \in C_x$, $j = 1, \ldots, |X|$.
3: Создать взаимно однозначную таблицу подстановки $E_Y : y^{(j)} \to c_y^{(j)}$, где $c_y^{(j)} \in C_y$, $j = 1, \ldots, |Y|$, $|Y|$ — мощность множества значений P, то есть количество уникальных $y^{(j)}$ в Y, $y^{(j)} = P(x)$, $x \in X$.
4: Выбрать архитектуру $\mathcal{N}$ с n входами, m выходами, не менее чем двумя слоями нейронов с нелинейной функцией активации и достаточным числом весовых коэффициентов.
5: Обучить нейронную сеть $\mathcal{N}$ на множестве пар $(c_x^{(j)}, c_y^{(j)})$, $j = 1, \ldots, |X|$, где $c_x^{(j)} = E_X(x^{(j)})$, $c_y^{(j)} = E_Y\big(P(x^{(j)})\big)$, так, чтобы

$$\forall c_x^{(j)} \in C_x \ \big(\hat{c}_y^{(j)} = \mathcal{N}(c_x^{(j)})\big) \Rightarrow \left|c_{yi}^{(j)} - \hat{c}_{yi}^{(j)}\right| < 0{,}5, \ i = 1, \ldots, m.$$

Алгоритм 2. Применение нейросетевого криптографического обфускатора

1: Передать обученную нейронную сеть $\mathcal{N}$ в недоверенное окружение.
2: Вычислить в доверенном окружении $c_x = E_X(x)$.
3: Передать c_x в недоверенное окружение.
4: Вычислить $c_y = \mathcal{N}(c_x)$ в недоверенном окружении.
5: Передать c_y в доверенное окружение.
6: Вычислить $y = D_Y(c_y)$ в доверенном окружении.

3. Шифрование данных нейросетевой криптографической обфускации

Рассмотрим способы, которые могут быть использованы для шифрования данных, подаваемых на вход нейросетевого криптографического обфускатора и получаемых в качестве результата его работы. Сравним стойкость шифрования с ключом и с помощью таблицы случайных взаимно однозначных подстановок.

Взаимно однозначные таблицы подстановки могут быть созданы с помощью симметричного блочного шифра с секретным ключом K. В таком случае $c_x = E_K(x)$ и $x = D_K(c_x)$. Поскольку обученная нейронная сеть $\mathcal{N}$ реализует обфускатор неразличимости, то информация о ключе шифрования не будет представлена в извлекаемом для злоумышленника виде. Стойкость шифра в этом случае определяется длиной ключа.

Таблицы подстановок могут быть также созданы с помощью генератора случайных чисел, и секретным ключом будет являться сама таблица целиком. Для практического использования в криптографических приложениях данный подход выглядит слишком ресурсоёмким по сравнению с коротким ключом длиной k бит, поскольку ключом выступает таблица размера $n \cdot 2^n$ бит, где n — длина шифруемого вектора. Однако с точки зрения ресурсов, затрачиваемых на обучение нейронной сети, эти подходы не отличаются, поскольку длительность обучения зависит от мощности (размера) обучающей выборки, то есть от размера таблицы, задающей число пар подстановок: $E_X : x^{(j)} \to c_x^{(j)}$, $j = 1, \ldots, |X|$.

Сравним стойкость шифрования булева вектора $x \in \mathbb{B}^n$ таблицей случайных подстановок со стойкостью шифрования с помощью ключа длиной k бит. Стойкость идеального блочного шифра определяется перебором всех возможных ключей, что для ключа длиной k бит составляет 2^k.

Для таблицы случайных подстановок количество вариантов ключа, то есть различных таблиц, составляет $(2^n)!$, то есть переборная стойкость определяется размерностью шифруемого вектора.

Наиболее распространённые современные блочные шифры имеют длину ключа $k = 256$ бит. Найдём, при каком размере таблицы случайных подстановок n достигается сравнимая переборная стойкость. Для этого решим неравенство $(2^n)! > 2^k$ для $k = 256$:

$$\log_2\bigl((2^n)!\bigr) > 256,$$

что даёт $117{,}6 < 256$ при $n = 5$ и $295{,}9 > 256$ при $n = 6$. Таким образом, при размере вектора $n \geqslant 6$ переборная стойкость шифра таблицы случайных подстановок превышает переборную стойкость современных шифров AES-256 и ГОСТ 34.12-2018 «Кузнечик» и «Магма».

Конечно, не все возможные таблицы случайных подстановок обеспечивают секретность шифрования, кроме того, некоторые из них реализуют тривиальные преобразования вроде шифра Цезаря и других, для которых найдены простые алгоритмы взлома. Следует также отметить, что для малых n можно успешно применять частотный анализ. Именно поэтому размер блока современных шифров составляет 128 бит.

Частотный анализ применим в предположении, что известно частотное распределение открытых данных. В случае нейросетевой обфускации прикладной программы P смысл зашифрованных данных для злоумышленника неизвестен, поскольку скрыта реализация алгоритма. Если априорно нет предположений о семантике обрабатываемых данных и назначении программы P, то ценность частотного анализа представляется незначительной.

Дополнительным усложнением задачи взлома для злоумышленника является возможность использования различных таблиц подстановок E_X и E_Y для входных и выходных данных. Таким образом, даже в случае тождественной программы $P : y = x$ шифрующие подстановки сделают взлом нетривиальным:

$$c_y = E_Y(D_X(c_x)).$$

К числу приёмов, позволяющих усложнить взлом, можно отнести увеличение размерности n входного вектора данных больше минимально необходимого для представления всего множества входных векторов X. Трудоёмкость процедуры обучения $\mathcal{N}$ при этом не вырастет, так как объём обучающей выборки останется таким же. Для того чтобы скрыть частотный состав шифруемых данных, можно с некоторой периодичностью подавать на вход нейросети случайные векторы c_x, в том числе не являющиеся результатом шифрования каких-либо допустимых $x \in X$. Выход нейросети $\mathcal{N}$ в этом случае не определён, и его тоже можно считать случайным.

Таким образом, в качестве подхода шифрования входных и выходных данных для нейросетевой криптографической обфускации целесообразно использовать таблицы случайных подстановок.

4. Обсуждение

Предложенный подход к реализации нейросетевого шифрования, совмещённого с обфускацией, обеспечивает реализацию функций, традиционно возлагаемых на го-

моморфное шифрование. В то же время нельзя сказать, что нейросетевая криптографическая обфускация может быть напрямую сопоставлена с гомоморфным шифрованием. Во-первых, гомоморфное шифрование основано на использовании ключа и его стойкость требует изучения, подобно всем криптографическим алгоритмам. В предложенном подходе алгоритм шифрования данных может быть произвольным, включая таблицы случайных подстановок, являющиеся идеальным блочным шифром [17].

Во-вторых, в гомоморфном шифровании алгоритм обработки зашифрованных данных не является секретом. Предложенный подход неделимо совмещает шифрование и обфускацию, защищая конфиденциальность как данных, так и алгоритма, который их обрабатывает.

Нейросетевая реализация криптографических алгоритмов уже находилась в фокусе внимания исследователей. Попытки синтеза криптографических систем на основе специфических свойств нейросетей, будь то хаотическая динамика [9, 10] или конкурентное обучение [11, 12], преследовали целью создание каких-то новых шифрующих алгоритмов. В противоположность этому, нейросетевая криптографическая обфускация не вводит нового алгоритма шифрования, предлагая использование существующих с ключом или на основе таблицы случайных подстановок.

Предлагаемый подход представляется фундаментальным и комплексным решением задачи защиты интеллектуальной собственности и конфиденциальных данных, частные подходы к которым с использованием искусственных нейронных сетей описаны ранее [13, 15].

Выводы

В работе предложен подход по нейросетевой криптографической обфускации алгоритмов, обрабатывающих зашифрованные данные. Настоящий механизм в целом преследует цели гомоморфного шифрования, но не эквивалентен ему. Представляется, что на основе этого подхода возможно реализовать доверенные вычисления в недоверенном вычислительном окружении, к которым относятся публичные современные облачные среды.

ЛИТЕРАТУРА

1. *Николенко С., Кадурин А., Архангельская Е.* Глубокое обучение. Погружение в мир нейронных сетей. СПб.: Питер, 2020.
2. *Алексеев Д. В.* Приближение функций нескольких переменных нейронными сетями // Фундаментальная и прикладная математика. 2009. Т. 15. № 3. С. 9–21.
3. *Хайкин С.* Нейронные сети: полный курс. 2-е изд. М.: Вильямс, 2008.
4. *Focardi R. and Luccio F. L.* Neural cryptanalysis of classical ciphers // Proc. ICTCS. 2018. http://ceur-ws.org/Vol-2243/paper10.pdf
5. *Danziger M. and Henriques M. A. A.* Improved cryptanalysis combining differential and artificial neural network schemes // Intern. Telecommun. Symp. (ITS). Sao Paulo, 2014. P. 1–5.
6. *Turcanik M.* Using recurrent neural network for hash function generation // Intern. Conf. Appl. Electronics (AE). Pilsen, 2017. P. 1–4.
7. *Lian S., Sun J., and Wang Z.* One-way hash function based on neural network // arXiv:0707.4032. 2007.
8. *Karras D. A. and Zorkadis V.* On neural network techniques in the secure management of communication systems through improving and quality assessing pseudorandom stream generators // Neural Networks. 2003. V. 16. Iss. 5–6. P. 899–905.

9. *Kanter I., Kinzel W., and Kanter E.* Secure exchange of information by synchronization of neural networks // Europhys. Lett. 2002. No. 57. P. 141–147.
10. *Klimov A., Mityagin A., and Shamir A.* Analysis of neural cryptography // LNCS. 2002. V. 2502. P. 288–298.
11. *Coutinho M., De Oliveira A. R., Borges F., et al.* Learning perfectly secure cryptography to protect communications with adversarial neural cryptography // Sensors. 2018. No. 18. Article 1306. https://pubmed.ncbi.nlm.nih.gov/29695066/
12. *Abadi M. and Andersen D. G.* Learning to protect communications with adversarial neural cryptography // arXiv:1610.06918. 2016.
13. *Xie P., Bilenko M., Finley T., et al.* Crypto-nets: Neural networks over encrypted data // arXiv:1412.6181. 2014.
14. *Hecht-Nielsen R.* Kolmogorov's mapping neural network existence theorem // IEEE First Annual Int. Conf. on Neural Networks, San Diego, 1987. V. 3. P. 11–13.
15. *Dathathri R., Saarikivi O., Chen H., et al.* CHET: an optimizing compiler for fully-homomorphic neural-network inferencing // Proc. PLDI 2019. N.Y.: ACM, 2019. P. 142–156.
16. *Елисеев В. Л.* Искусственные нейронные сети как механизм обфускации вычислений // Прикладная дискретная математика. Приложение. 2019. № 12. С. 165–169.
17. *Фергюсон Н., Шнайер Б.* Практическая криптография. М.: Диалектика, 2005.

УДК 004.75 DOI 10.17223/2226308X/13/26

МЕТОД СОКРЫТИЯ ПРИВАТНЫХ ДАННЫХ ДЛЯ БЛОКЧЕЙН-СИСТЕМЫ ПРОВЕДЕНИЯ ТЕНДЕРОВ[1]

Д. О. Кондырев

Предложен новый метод, позволяющий решить проблему приватности информации в открытых блокчейн-системах с использованием криптографического протокола доказательства с нулевым разглашением zk-SNARK. Метод реализован в виде криптографической схемы на основе библиотеки libsnark и интегрирован в модифицированный Ethereum C++ клиент.

Ключевые слова: *тендеры, распределённые системы, блокчейн, доказательство с нулевым разглашением, zk-SNARK, платформа Ethereum.*

На сегодняшний день большинство конкурсных закупок и электронных торгов проводятся через специализированные информационные системы. В таких системах участники должны быть уверены в том, что никто не имеет возможности нарушить правила проведения тендера или получить доступ к конфиденциальной информации. Решить проблему доверия при проведении тендеров позволяет блокчейн. Однако при использовании этой технологии все данные сохраняются в открытом виде и доступны всем участникам. В случае с тендерами открытость информации нарушает тайну заявок, которая должна быть сохранена до окончания этапа запроса предложений.

Ранее была разработана блокчейн-система для проведения тендеров с шифрованием заявок [1]. Однако такой подход не позволяет проверить корректность зашифрованной заявки в момент её подачи. Ещё одним недостатком является то, что все участники могут наблюдать факт подачи заявки пользователем.

[1]Работа выполнена при поддержке Математического центра в Академгородке (г. Новосибирск), соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2019-1613, и Лаборатории криптографии JetBrains Research.

В данной работе предложена и реализована система тендеров, которая удовлетворяет критериям безопасности, открытости и конфиденциальности. Вопрос доверия решён с помощью технологии блокчейн, а сокрытие приватной информации — с помощью криптографического протокола неинтерактивного доказательства знания с нулевым разглашением zk-SNARK [2]. Система основана на платформе Ethereum. Вся ключевая информация о тендерах сохраняется в блокчейне, а проверка правил и отслеживание выполнения условий участниками реализованы в виде кода смарт-контрактов.

Для реализации алгоритма сокрытия информации о заявках в Ethereum C++ клиент добавлен отдельный модуль *tenderzkp*. Он построен на базе протокола zk-SNARK с предобработкой для NP-полного языка системы ограничений ранга 1. Протокол использует эллиптическую кривую Баррето — Наерига. Реализация криптографической схемы предоставлена библиотекой libsnark [3].

В модуле *tenderzkp* реализованы функции создания и верификации доказательства о корректности заявки. Доказательство строится на основе ограничений на приватные и открытые входные данные заявки, выраженных с помощью базовых схем библиотеки libsnark.

Для работы с добавленной криптографической схемой в Ethereum C++ клиент созданы новые предкомпилированные контракты с адресами 0x00...09 и 0x00...0a и разработана Solidity-библиотека, которая инкапсулирует низкоуровневое взаимодействие с предкомпилированными контрактами и предоставляет интерфейс для работы с ними в виде Solidity-функций. Чтобы добавить возможность вызывать методы разработанной криптографической схемы из сторонних приложений, расширен JSON-RPC API Ethereum клиента.

Предложенный метод может быть использован не только для тендеров, но и в других системах, где есть необходимость скрывать часть информации в открытой блокчейн-сети. Он расширяет область применения технологии блокчейн в промышленных программных комплексах.

ЛИТЕРАТУРА

1. *Hardwick F. S., Akram R. N., and Markantonakis K.* Fair and transparent blockchain based tendering framework — A step towards open governance // IEEE Intern. Conf. TrustCom/BigDataSE, New York, USA, 2018. P. 1342–1347.
2. *Ben-Sasson E., Chiesa A., Genkin D., et al.* SNARKs for C: Verifying program executions succinctly and in zero knowledge // CRYPTO'2013. LNCS. 2013. V. 8043. P. 90–108.
3. `https://github.com/scipr-lab/libsnark` — libsnark: a C++ library for zkSNARK proofs.

UDC 004.056 DOI 10.17223/2226308X/13/27

VALIDATION-FREE OFFCHAIN TRANSACTIONS WITH UNLINKABLE DOUBLE SPEND DETECTION

S. N. Kyazhin, K. A. Klimenko

The so-called layer-two protocols are a class of blockchain scaling solutions. They allow to minimize onchain traffic, and therefore make state transitions (payments, for example) faster and more suitable for everyday use, while still preventing double spend attacks. Unfortunately, these solutions also have some downsides and tradeoffs (channel capacity, route availability, operator availability, etc.). In this work we study the possibility of simplifying and improving existing protocols for offchain transactions and describe a scheme that, without transaction validation, allows to detect a double

spender and not trace other transactions. This scheme is based on the anonymous transferable e-cash system. We use an offchain analogue of the UTXO model, therefore there are offchain transactions for issue, transfer and redeem of a so-called note, containing a number that can be used as a secret key to make the corresponding token transfer transaction onchain.

Ключевые слова: *blockchain, offchain, unlinkability, double spend detection.*

1. Introduction

Layer-two protocols are the trend of blockchain scalability solutions right now. Such protocols allow users to make offchain transactions. In [1] the authors summarize and systematize existing solutions: payment/state channels and commit-chains (or hubs). It would be interesting to create a system in which some users transfer tokens to others, while they can dynamically join the two-layer solution (free establishment property [1]). Next, any user who received tokens offchain can receive them onchain. There are commit-chains with unlinkability and anonimity properties (e.g. TumbleBit [2] and Bolt [3]), which are suitable for this problem.

However, existing solutions also have the following tradeoffs. For example:

— unlike the regular blockchain security model, when using a layer-two solution, the user may need to monitor his funds from time to time;
— there may be some constraints or prerequisites for using such a solution (channel capacity, route availability for payment/state channels, operator availability for commit-chains, etc.);
— the user may need to use additional complex software that stores sensitive information (the history of transactions, including the so-called “breach remedy transactions”, or other data required to create a proof of fraud and prevent loss of funds).

Consider the following case: there is no transaction validation, however, there is an operator that checks for double spend when the current owner wants to receive his tokens in the blockchain. Obviously, in this case, the operator will not be able to prevent double spend, but can only detect it.

This case is very similar to an e-cash system. Moreover, for example, Bolt uses an offline anonymous e-cash scheme [4]. But this protocol does not have the transferability property (for a coin there can be only one transfer transaction). The paper [5] describes a modified protocol that provides transferability (the main modification is related to the ability for the receiver to spend the received coin later).

Our current research aims to implement the approaches proposed in [4, 5] to create a simpler offchain transaction scheme that allows to detect a double spender without validation and linking transactions.

Let $\mathbb{G}$ be an additive group of prime order q, $s \in \mathbb{Z}_q$ be a number that can be used as a secret key to make a transaction for transfer of some tokens onchain. All other blockchain details are beyond the scope of this paper.

Suppose we have an offchain analogue of the UTXO model. For each s there is a so-called note. Therefore, an offchain token transfer transaction means a transaction for transfer of a note with s value.

Let G, H be generators in $\mathbb{G}$, $x_i \in \mathbb{Z}_q$ ($P_i = x_i G$) be the private (public) key of the i-th participant of the offchain transaction scheme, $i = 1, \ldots, n$. We formulate the problem as follows — to create a scheme based on [5] that allows:

— to transfer of a note between users with these keys without transaction validation;

— to reveal the public key of the user who transfered the note more than once when the current owner of the note wants to make the corresponding transaction onchain and not link the transactions for the note transfer with the corresponding public keys of other users.

2. Description of the Scheme

Let $\mathcal{H}$ be a cryptographic hash function to $\mathbb{Z}_q$, $\pi(\{a_1,\dots,a_n\},\{b_1,\dots,b_n\} : f_1(a_1,\dots,a_n, b_1,\dots,b_n) = c_1,\dots,f_n(a_1,\dots,a_n,b_1,\dots,b_n) = c_n)$ be a zero knowledge proof of knowledge of such private $a_1,\dots,a_n$ and public $b_1,\dots,b_n$ (in general from different sets) that $f_1(a_1,\dots,a_n,b_1,\dots,b_n) = c_1,\dots,f_n(a_1,\dots,a_n,b_1,\dots,b_n) = c_n$, where $f_1,\dots,f_n$ are the corresponding functions, $c_1,\dots,c_n$ are constants.

2.1. Note Issue

To issue a note, the owner of the tokens in the blockchain (he knows s):

— generates some message msg (transaction description) and computes the hash function $m_0 = \mathcal{H}(msg)$;
— computes $r_0 = (x + m_0)^{-1}G$, where x is the owner's private key;
— computes $\tilde{r}_0 = \mathcal{H}(r_0, m_0)$;
— computes $T_0 = xG + \tilde{r}_0(x + s + 1)^{-1}H$;
— computes a proof

$$\pi_0 = \pi(\{x\},\{s, T_0, r_0, m_0\} : T_0 = xG + \tilde{r}_0(x + s + 1)^{-1}H, r_0 = (x + m_0)^{-1}G);$$

— creates the note $(s,\, V_0)$, where $V_0 = (T_0, \pi_0, r_0, m_0)$.

2.2. Note Transfer

Assume that user A owns a note $(s,\, V)$, where $V = (V_0,\dots,V_l)$, $V_j = (T_j, \pi_j, r_j, m_j)$, $j = 0,\dots,l$, that he legitimately received from another user. If A legitimately received the note $(s,\, V)$, it is necessary that $r_l = (x_A + m_l)^{-1}G$, where x_A is the private key of A.

The following steps describe the interactive procedure for transfer of the note $(s,\, V)$ from user A to user B.

First, the Receiver (B):

— generates some message msg (transaction description) and computes the hash function $m_{l+1} = \mathcal{H}(msg)$;
— computes $r_{l+1} = (x_B + m_{l+1})^{-1}G$, where x_B is the receiver's private key;
— sends m_{l+1}, r_{l+1} to the Sender.

Next, the Sender (A):

— computes $\tilde{r}_{l+1} = \mathcal{H}(r_{l+1}, m_{l+1})$;
— computes $h_{l+1} = \mathcal{H}(s, T_0,\dots,T_l)$;
— computes $T_{l+1} = x_AG + \tilde{r}_{l+1}(x_A + s + h_{l+1})^{-1}H$;
— computes a proof

$$\pi_{l+1}{=}\pi(\{x_A\},\{s, T_{l+1}, r_{l+1}, m_{l+1}\}{:}T_{l+1} = x_AG{+}\tilde{r}_{l+1}(x_A{+}s{+}h_{l+1})^{-1}H, r_l = (x_A{+}m_l)^{-1}G);$$

— creates and sends the note (s, V), where $V = (V_0,\dots,V_{l+1})$, $V_{l+1} = (T_{l+1}, \pi_{l+1}, r_{l+1}, m_{l+1})$, to the Receiver.

The Receiver can optionally verify the proof π_{l+1}.

2.3. Note Redeem

When the current owner of the note wants to make the corresponding transaction in the blockchain, he sends the note (s, V), $V = (V_0, \ldots, V_t)$, $V_j = (T_j, \pi_j, r_j, m_j)$, to the Operator.

The Operator verifies that:

— the proof π_j is valid for all $j = 0, \ldots, t$;
— the note with s has not been redeemed.

If the note with s has been redeemed, there was a double spend.

2.4. Double Spender Detection

A double spend is equivalent to the fact that the Operator received notes with the same s and different $V = (V_0, \ldots, V_k, \ldots, V_t)$ and $V' = (V_0, \ldots, V'_k, \ldots, V'_{t'})$.

The Operator:

— looks for the minimal k that $V_k = (T_k, \pi_k, r_k, m_k) \neq V'_k = (T'_k, \pi'_k, r'_k, m'_k)$;
— computes $\tilde{r}_k = \mathcal{H}(r_k, m_k)$ and $\tilde{r}'_k = \mathcal{H}(r'_k, m'_k)$;
— computes the public key of the double spender:

$$P = (\tilde{r}'_k - \tilde{r}_k)^{-1}(\tilde{r}'_k T_k - \tilde{r}_k T'_k).$$

3. Conclusion

This paper is dedicated to the research of the possibility of constructing a protocol for offchain transactions that, without transaction validation, allows to detect a double spender and not trace other transactions. We describe a possible scheme based on the transferable anonymous e-cash system proposed in [5]. In future papers, we would like to reformulate the security properties from [5] and provide the proofs.

REFERENCES

1. *Gudgeon L., Moreno-Sanchez P., Roos S., et al.* SoK: Off The Chain Transactions. Cryptology ePrint Archive: Report 2019/360.
2. *Heilman E., Alshenibr L., Baldimtsi F., et al.* TumbleBit: An Untrusted Bitcoin-Compatible Anonymous Payment Hub. Cryptology ePrint Archive: Report 2016/575.
3. *Green M. and Miers I.* Bolt: Anonymous Payment Channels for Decentralized Currencies. Cryptology ePrint Archive: Report 2016/701.
4. *Camenisch J., Hohenberger S., and Lysyanskaya A.* Compact E-Cash // EUROCRYPT 2005. LNCS. 2005. V. 3494. P. 302–321.
5. *Canard S., Gouget A., and Traore J.* Improvement of efficiency in (unconditional) anonymous transferable E-Cash // Financial Cryptography and Data Security. LNCS. 2008. V. 5143. P. 202–214.

Секция 5

ПРИКЛАДНАЯ ТЕОРИЯ КОДИРОВАНИЯ, АВТОМАТОВ И ГРАФОВ

УДК 003.26, 519.725, 519.176 DOI 10.17223/2226308X/13/28

О НОВЫХ ОЦЕНКАХ РАЗМЕРНОСТИ ПОДКОДОВ КОДОВ РИДА — МАЛЛЕРА, КВАДРАТ АДАМАРА КОТОРЫХ МАКСИМАЛЕН

В. В. Высоцкая

Наличие в коде некоторой структуры может привести к снижению стойкости всей системы, построенной на нем. Для «маскировки» кода под код «общего вида» часто используются подкоды. Однако стойкость подкодов, квадрат Адамара которых равен квадрату полного кода, сводится к стойкости этого кода. Таким образом, данное свойство необходимо учитывать как при синтезе схем на кодах, так и при их криптоанализе. В работе анализируется минимальное количество мономов степени r, которые при добавлении к коду $RM(r-1,m)$ образуют подкод, квадрат Адамара которого максимален, т. е. совпадает с кодом $RM(2r,m)$. Это число снизу оценивается аналитически, а для получения верхней оценки предлагается жадный алгоритм построения такого набора мономов.

Ключевые слова: *постквантовая криптография, кодовая криптография, коды Рида — Маллера, подкоды Рида — Маллера, произведение Адамара, криптосистема Мак-Элиса.*

В последнее время большую популярность получили кодовые криптосистемы. Этот интерес прослеживается в работах, поданных на конкурс на перспективный постквантовый алгоритм, объявленный NIST [1] в 2016 г. для дальнейшей стандартизации. Кроме того, ТК 26 выбрал схемы на кодах как одно из направлений разработки будущего российского стандарта постквантовых алгоритмов.

При синтезе новых кодовых схем одним из самых важных вопросов является выбор базового кода, от которого будут зависеть все характеристики. В целях создания асимметрии в возможностях легального пользователя и противника требуется скрывать структуру кода. Это можно сделать, например, используя подкоды. Однако в работе И. В. Чижова и М. А. Бородина [2] стойкость криптосистемы Мак-Элиса [3] на подкодах коразмерности 1 сведена к стойкости оригинальной криптосистемы. Сведение работает для подкодов, квадрат Адамара которых совпадает с квадратом базового кода. Такое свойство подкодов без наложения ограничений на коразмерность исследовано в работе [4].

Определение 1. *Кодом Рида — Маллера* $RM(r,m)$ называется множество булевых функций f от m переменных, таких, что $\deg(f) \leqslant r$.

Определение 2. *Произведением Адамара двух векторов* называется вектор, полученный в результате покомпонентного произведения координат этих векторов:

$$(a_1,\ldots,a_n)\circ(b_1,\ldots,b_n)=(a_1b_1,\ldots,a_nb_n),$$

а *произведение Адамара двух кодов* $\mathcal{A}$ *и* $\mathcal{B}$ есть линейная оболочка всех попарных произведений вида $a\circ b$, где $a\in\mathcal{A}$, $b\in\mathcal{B}$.

Отметим, что квадрат Адамара кода Рида — Маллера имеет смысл только при $m \geqslant 2r$.

Будем рассматривать подкоды вида

$$\big(RM(r-1,m) \cup \{f_1, \ldots, f_{w(m,r)}\}\big)^2 = RM(2r,m), \tag{1}$$

где $f_1, \ldots, f_{w(m,r)}$ — мономы степени r, а под возведением в квадрат понимается квадрат Адамара. Задача состоит в минимизации значения $w(m,r)$.

Перейдём к графовой интерпретации задачи. Сопоставим подкод $\mathcal{A} \subset RM(r,m)$ с гиперграфом G с m вершинами, помеченными как $x_1, \ldots, x_m$. Ребро $\{x_{i_1}, \ldots, x_{i_r}\}$ проведено тогда и только тогда, когда моном $x_{i_1} \ldots x_{i_r} \in \mathcal{A}$. В [4] показано, что для обеспечения условия (1) достаточно, чтобы гиперграф был стабильным.

Определение 3. Гиперграф называется *стабильным*, если каждое множество, состоящее из $2r$ вершин, покрыто двумя непересекающимися r-рёбрами.

Очевидно, что задача поиска стабильного гиперграфа с минимальным количеством r-рёбер эквивалентна поиску минимального числа $w(m,r)$. В работе [4] это число для $r \geqslant 2$ и $h < r/3$ оценено как

$$\mathrm{C}_m^{2r}/\mathrm{C}_{m-r}^r \leqslant w(m,r) \leqslant \mathrm{C}_m^r - T(r,m,h)\,(\mathrm{C}_{2r}^r - 2)\,, \tag{2}$$

где

$$T(r,m,h) = \max\big\{t : \exists S_1, \ldots, S_t \big(S_i \subset \{1,\ldots,m\} \;\&\; \\ \&\; |S_i| = 2r \;\&\; (i \neq j \Rightarrow |S_i \cap S_j| \leqslant h),\; i,j \in \{1,\ldots,t\}\big)\big\}.$$

Эти оценки могут быть улучшены.

Теорема 1.

$$w(m,r) \geqslant \sqrt{\gamma + 2\mathrm{C}_m^{2r}} + \sqrt{\gamma}, \quad \text{где} \quad \gamma = \sum_{i=\max\{1,3r-m\}}^{r-1} \mathrm{C}_r^i.$$

Для получения верхней оценки предложен алгоритм, который по параметрам кода строит соответствующий стабильный гиперграф. Его реализацию, написанную на Python, можно посмотреть в `https://github.com/VysotskayaVictory/StableGraphGreedy/`. На каждом шаге алгоритма происходит попытка добавить новое r-ребро. В случае успеха значение $w(m,r)$ увеличивается на 1, а также формируется список r-рёбер, которые пересекаются с данным. Они добавляются в список `inter`. Вместе с этим поддерживается счетчик `repeated` повторно покрытых множеств. Алгоритм останавливается, когда все C_m^{2r} множеств размера $2r$ покрыты парами непересекающихся r-рёбер. Условием останова является выполнение равенства

$$\mathrm{C}_{w(m,r)}^2 - |\texttt{inter}| - \texttt{repeated} = \mathrm{C}_m^{2r}.$$

Сравнение полученных оценок с оценками из (2) представлено на рис. 1. Его анализ позволяет говорить о том, что оценки существенно улучшены.

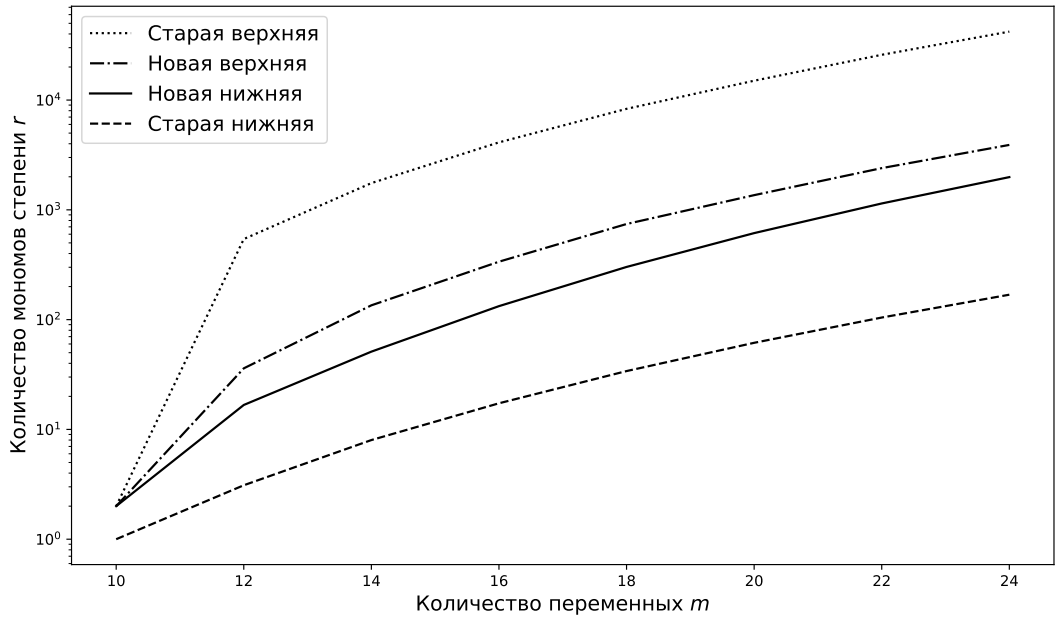

Рис. 1

ЛИТЕРАТУРА

1. `https://csrc.nist.gov/Projects/post-quantum-cryptography/Post-Quantum-Cryptography-Standardization/Call-for-Proposals`.
2. *Чижов И. В., Бородин М. А.* Классификация произведений Адамара подкодов коразмерности 1 кодов Рида — Маллера // Дискретная математика. 2020. № 32(1). С. 115–134.
3. *McEliece R. J.* A public-key cryptosystem based on algebraic coding theory // DSN Progress Report. 1978. No. 4244. P. 114–116.
4. *Vysotskaya V. V.* Characteristics of Hadamard Square of Reed — Muller Subcodes of Special Type. `https://eprint.iacr.org/2020/507`.

УДК 519.1 DOI 10.17223/2226308X/13/29

О КОЛИЧЕСТВЕ НЕДОСТИЖИМЫХ СОСТОЯНИЙ В КОНЕЧНЫХ ДИНАМИЧЕСКИХ СИСТЕМАХ ОРИЕНТАЦИЙ ПОЛНЫХ ГРАФОВ

А. В. Жаркова

Рассматриваются конечные динамические системы ориентаций полных графов. Состояниями системы являются все возможные ориентации полного графа, а эволюционная функция задаётся следующим образом: динамическим образом данного орграфа является орграф, полученный из исходного путём переориентации всех дуг, входящих в стоки, других отличий между исходным орграфом и его образом нет. Приводятся формулы для подсчёта количества недостижимых и достижимых состояний в рассматриваемых системах, представлены соответствующие таблицы для полных графов с количеством вершин от двух до десяти.

Ключевые слова: *граф, достижимое состояние, источник, конечная динамическая система, недостижимое состояние, ориентация графа, полный граф, сток, турнир, эволюционная функция.*

Графовые модели, в которых отказы процессоров интерпретируются как удаление соответствующих вершин, а отказы сетевых каналов — как удаление дуг, занимают важное место в задачах, связанных с отказоустойчивостью компьютерных сетей. При изучении модельных графов можно применять идеи и методы теории конечных динамических систем [1 – 3]. В модели [1] в качестве механизма восстановления работоспособности сети предлагается так называемая SER-динамика бесконтурных связных ориентированных графов. В настоящей работе полные графы изучаются с точки зрения динамического подхода к отказоустойчивости графовых систем.

Под *ориентированным графом* (*орграфом*) понимается пара $\overrightarrow{G} = (V, \beta)$, где V — конечное непустое множество вершин; $\beta \subseteq V \times V$ — отношение смежности на множестве V (пара $(u, v) \in \beta$ называется *дугой* орграфа). *Неориентированным графом* (или, для краткости, *графом*) называется пара $G = (V, \beta)$, где β — симметричное и антирефлексивное отношение на множестве вершин V. Дуги неориентированного графа называют *рёбрами*. Орграф $\overrightarrow{G} = (V, \beta)$ называется *направленным графом* (или *дографом*), если отношение β антисимметрично. Граф $G = (V, \beta)$ называется *полным*, если любые две его вершины соединены ребром. Полный граф с n вершинами обозначим K_n. *Турниром* называется полный направленный граф. Говорят, что вершина v *достижима* из вершины u, если в орграфе существует путь из u в v. Вершина орграфа, не достижимая из других его вершин, называется *источником*, а вершина, из которой не достижима никакая другая вершина, — *стоком* [4].

Под *конечной динамической системой* понимается пара (S, δ), где S — конечное непустое множество *состояний системы*, $\delta : S \to S$ — отображение множества состояний в себя, называемое *эволюционной функцией системы*. Каждой конечной динамической системе сопоставляется карта, представляющая собой орграф с множеством вершин S и дугами, проведёнными из каждой вершины $s \in S$ в вершину $\delta(s)$. Компоненты связности графа, задающего динамическую систему, называются её *бассейнами*.

Основными проблемами теории конечных динамических систем являются задачи отыскания эволюционных параметров системы без проведения динамики. К их числу относятся *ветвление* (количество непосредственных предшественников данного состояния) и, в частности, свойство *недостижимости* состояния (то есть когда состояние имеет нулевое ветвление). Автором описаны недостижимые состояния конечных динамических систем всех возможных ориентаций графов [5], подсчитаны количества недостижимых состояний в системах, связанных с ориентациями цепей, циклов, пальм [6]. В данной работе предлагаются формулы для подсчёта количества недостижимых и количества достижимых состояний в конечных динамических системах ориентаций полных графов.

Пусть дан полный граф K_n, $n > 1$, $m = n(n-1)/2$ — число рёбер. Придадим его рёбрам произвольную ориентацию, тем самым получив направленный граф (турнир) $\overrightarrow{G}$. Применим к полученному орграфу эволюционную функцию α, которая у данного орграфа одновременно переориентирует все дуги, входящие в стоки, а остальные дуги оставляет без изменения, в результате чего получим орграф $\alpha(\overrightarrow{G})$, других отличий между $\overrightarrow{G}$ и $\alpha(\overrightarrow{G})$ нет. Если проделать указанные действия со всеми возможными ориентациями данного графа, то получим карту конечной динамической системы (Γ_{K_n}, α), $n > 1$, где через Γ_{K_n} обозначим множество всех возможных ориентаций данного полного графа K_n, $|\Gamma_{K_n}| = 2^m$, состоящую из одного или нескольких бассейнов.

На рис. 1 изображена карта конечной динамической системы (Γ_{K_3}, α).

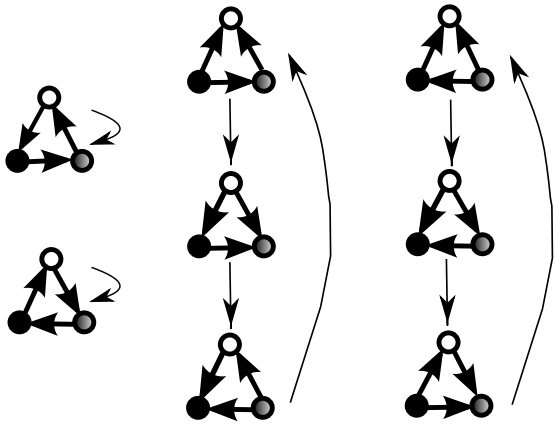

Рис. 1. Карта конечной динамической системы (Γ_{K_3}, α)

Теорема 1. Состояние $\overrightarrow{G} \in \Gamma_{K_n}$ конечной динамической системы (Γ_{K_n}, α), $n > 1$, недостижимо тогда и только тогда, когда в орграфе $\overrightarrow{G}$ нет источника и есть сток.

Теорема 2. Количество недостижимых состояний в конечной динамической системе (Γ_{K_n}, α), $n > 1$, равно

$$\text{КНС}_{(\Gamma_{K_n},\alpha)} = n\big(2^{(n-1)(n-2)/2} - (n-1)2^{(n-2)(n-3)/2}\big).$$

Следствие 1. Количество достижимых состояний в конечной динамической системе (Γ_{K_n}, α), $n > 1$, равно

$$\text{КДС}_{(\Gamma_{K_n},\alpha)} = 2^{n(n-1)/2} - n\big(2^{(n-1)(n-2)/2} - (n-1)2^{(n-2)(n-3)/2}\big).$$

В таблице приведены данные по количеству недостижимых и достижимых состояний в конечных динамических системах (Γ_{K_n}, α) для $2 \leqslant n \leqslant 10$, полученные с помощью вычислительных экспериментов.

n	$\text{КНС}_{(\Gamma_{K_n},\alpha)}$	$\text{КДС}_{(\Gamma_{K_n},\alpha)}$
2	0	2
3	0	8
4	8	56
5	160	864
6	4224	28544
7	186368	1910784
8	14942208	253493248
9	2264924160	66454552576
10	663035576320	34521336512512

Можно заметить, что в конечных динамических системах (Γ_{K_n}, α) большинство состояний являются достижимыми.

ЛИТЕРАТУРА

1. *Barbosa V. C.* An atlas of edge-reversal dynamics. London: Chapman&Hall/CRC, 2001.
2. *Khrennikov A. and Nilsson M.* On the number of cycles of p-adic dynamical systems // J. Number Theory. 2001. V. 90. P. 255–264.

3. *Салий В. Н.* Об одном классе конечных динамических систем // Вестник Томского госуниверситета. Приложение. 2005. № 14. С. 23–26.
4. *Богомолов А. М., Салий В. Н.* Алгебраические основы теории дискретных систем. М.: Наука, 1997.
5. *Жаркова А. В.* О ветвлении и непосредственных предшественниках состояний в конечной динамической системе всех возможных ориентаций графа // Прикладная дискретная математика. Приложение. 2013. № 6. С. 76–78.
6. *Жаркова А. В.* Недостижимые состояния в динамических системах, ассоциированных с цепями и циклами // Изв. Сарат. ун-та. Нов. сер. Сер. Математика. Механика. Информатика. 2011. Т. 11. Вып. 4. С. 116–123.

УДК 519.17 DOI 10.17223/2226308X/13/30

ОБ ОПТИМАЛЬНОСТИ РЕАЛИЗАЦИЙ ГРАФОВ С ЗАДАННЫМИ МЕРАМИ СВЯЗНОСТИ[1]

Б. А. Теребин, М. Б. Абросимов

Две основные меры связности графа — вершинная k и рёберная λ — связаны с минимальной степенью вершины δ графа известным соотношением Уитни: $k \leqslant \lambda \leqslant \delta$. Г. Чартрэнд и Ф. Харари доказали, что этот результат не улучшаем в том смысле, что для любых натуральных чисел a, b, c, таких, что $a \leqslant b \leqslant c$, можно построить граф, у которого $k = a$, $\lambda = b$, $\delta = c$. В доказательстве Чартрэнда и Харари предлагается граф с числом вершин $2(c+1)$ и числом рёбер $c(c+1)+b$. В данной работе рассматривается вопрос построения соответствующей реализации с наименьшим возможным числом вершин и рёбер.

Ключевые слова: *вершинная связность, рёберная связность, неравенство Уитни.*

1. Условие Уитни

Связные графы имеют важнейшее значение с точки зрения прикладной теории графов. Две основные меры связности графа — вершинная k и рёберная λ. *Вершинной связностью* k графа G называется наименьшее число вершин, удаление которых приводит к несвязному или тривиальному графу. *Рёберная связность* λ графа G определяется как наименьшее количество рёбер, удаление которых приводит к несвязному или тривиальному графу. Основные определения используются по работе [1].

Вершинная связность графа k, его рёберная связность λ и минимальная степень вершины δ связаны неравенством Уитни [2]:

Теорема 1 [2]. Для любого графа G справедливо неравенство $k \leqslant \lambda \leqslant \delta$.

Результат теоремы является неулучшаемым:

Теорема 2 (Чартрэнд, Харари [3]). Для любых натуральных чисел a, b, c, таких, что $a \leqslant b \leqslant c$, существует граф G, у которого $k = a$, $\lambda = b$, $c = \delta$.

Из доказательства теоремы 2 следует, что для любых a, b, c можно построить граф с числом вершин $2(c + 1)$ и числом рёбер $c(c + 1) + b$. Предлагается схема построения соответствующего графа: необходимо взять два полных графа K_{c+1}, в одном выбрать a вершин, в другом — b вершин и соединить выбранные вершины b рёбрами.

[1]Работа выполнена при поддержке Минобрнауки России в рамках выполнения государственного задания (проект № FSRR-2020-0006).

Возникает вопрос: можно ли для заданных $k = a$, $\lambda = b$, $c = \delta$ построить граф с меньшим числом вершин и рёбер? Оказывается, что в некоторых случаях это действительно возможно, что и является предметом исследования данной работы.

2. Основные результаты

Результат теоремы 2 можно улучшить не всегда, поэтому общий случай $a \leqslant b \leqslant c$ разделим на следующие неравенства:

1) $a \leqslant b < c$;
2) $a = b = c$;
3) $a < b = c$.

Для первого случая оказалось, что результат теоремы 2 является оптимальным по числу вершин:

Теорема 3. Граф с наименьшим количеством вершин, удовлетворяющий условию Уитни при $a \leqslant b < c$, является графом с числом вершин $2(c + 1)$.

Однако число рёбер может быть меньше. Следующий пример иллюстрирует данный случай.

Пример 1. Пусть $a = b = 4$, $c = 5$, т. е. $k = \lambda = 4$, $\delta = 5$.

Количество вершин равно $2(c + 1) = 2(5 + 1) = 12$. Количество рёбер в реализации из теоремы 2 должно быть $c(c + 1) + b = 34$. На рис. 1 изображён граф, построенный по теореме 3, с числом рёбер 30.

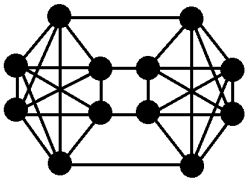

Рис. 1

В двух остальных случаях результат теоремы 2 может быть улучшен и по числу вершин. Второй случай является достаточно тривиальным:

Теорема 4. Граф с наименьшим количеством вершин, удовлетворяющий условию Уитни при $a = b = c$, является полным графом с числом вершин $c + 1$.

Наиболее интересным оказался третий случай, для которого удалось получить следующий результат:

Теорема 5. Граф с наименьшим количеством вершин, удовлетворяющий условию Уитни при $a < b = c$, является графом с числом вершин $2(c + 1) - a$.

Доказательство теоремы также является конструктивным и предлагает схему построения соответствующего графа. Следует отметить, что в общем случае можно построить несколько реализаций с числом вершин, как в теореме 5, но с разным числом рёбер. Схема из теоремы 5 позволяет строить граф не только с минимальным числом вершин, но и с минимально возможным числом рёбер.

Теорема 6. Граф с наименьшим количеством рёбер, удовлетворяющий условию Уитни при $a < b = c$, является графом с числом рёбер $c^2 - a^2 + a + c + \sigma$, где

$$\sigma = \begin{cases} 0, & \text{если } \lceil (2a^2 - ac - 2a)/2 \rceil \leqslant 0, \\ \lceil (2a^2 - ac - 2a)/2 \rceil & \text{иначе.} \end{cases}$$

Следующий пример иллюстрирует этот случай.

Пример 2. Пусть $a = 4$, $b = c = 5$, т. е. $k = 4$, $\lambda = \delta = 5$.

Данный случай удовлетворяет условию теорем 5 и 6. Согласно теореме 5, количество вершин равно $2(c + 1) - a = 2(5 + 1) - 4 = 8$. Найдем значение σ:

$$\sigma = \lceil (2a^2 - ac - 2a)/2 \rceil = \lceil (36 - 20 - 8)/2 \rceil = 2.$$

Тогда число рёбер равно $c^2 - a^2 + a + c + \sigma = 25 - 16 + 4 + 5 + 2 = 20$. На рис. 2 изображён соответствующий граф.

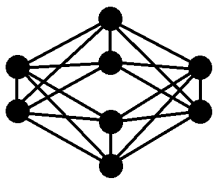

Рис. 2

ЛИТЕРАТУРА

1. *Харари Ф.* Теория графов. М.: Мир, 1973. 300 с.
2. *Whitney H.* Congruent graphs and the connectivity of graphs // Am. J. Math. 1932. V. 54. P. 150–168.
3. *Chartrand G. and Harary F.* Graphs with prescribed connectivities // 1966 Symp. on Graph Theory. Tihany, Acad. Sci. Hung. 1967. P. 61–63.

Секция 6

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

УДК 519.682 DOI 10.17223/2226308X/13/31

ГЕОМЕТРИЧЕСКОЕ УСЛОВИЕ РАЗРЕШИМОСТИ ФОРМАЛЬНЫХ ГРАММАТИК

О. И. Егорушкин, И. В. Колбасина, К. В. Сафонов

В работе продолжено развитие метода исследований формальных грамматик, под которыми подразумеваются системы некоммутативных полиномиальных уравнений. Такие системы решаются в виде формальных степенных рядов (ФСР), которые выражают нетерминальные символы алфавита через терминальные; первая компонента решения является формальным языком. Метод, развиваемый авторами, основывается на изучении коммутативного образа грамматики и формального языка, а именно: всякому ФСР поставлен в соответствие его коммутативный образ, который получается, если считать, что все символы являются коммутативными переменными. Получена теорема, которая даёт достаточное геометрическое условие того, что формальная грамматика имеет единственное решение в виде ФСР.

Ключевые слова: *системы полиномиальных уравнений, некоммутативные переменные, формальный степенной ряд, коммутативный образ, аналитичекая гиперповерхность.*

Как известно, теория формальных языков имеет фундаментальное значение не только для лингвистики, но и программирования. На ней базируются поиск в интернете, машинный перевод текстов, речевой диалог с компьютером, развитие языка математических абстракций, распознавание генетического кода в биоинформатике, обнаружение кода мигрирующих вирусов, разработка и реализация языков программирования, оптимизация компиляторов, перенос разработанных компьютерных программ в новую вычислительную среду и другие современные технологии.

Все эти приложения используют взаимосвязи языка (как множества возможных текстов) с грамматикой (сводом формальных правил, определяющих языковые конструкции и их равнозначимость). Более того, всюду нужны быстрые качественные алгоритмы формальных построений грамматики по языку и языка по грамматике и синтаксического анализа конструкций, немыслимые без серьёзного теоретического обоснования.

Контекстно-свободные грамматики, наряду с регулярными выражениями, активно используются для решения задач, связанных с разработкой формальных языков и синтаксических анализаторов [1 – 3]. Одним из основных достоинств контекстно-свободных грамматик является возможность задания широкого класса языков при сохранении относительной компактности представления [4, 5]. С уровнем контекстно-свободных языков соотносится атрибутивная модель процесса, определяющая описание моделируемого процесса в виде моделей бизнес-процессов. Переход к контекстно-свободным языкам выполняется за счёт моделирования бизнес-процесса с применени-

ем инструментов структурного или объектно-ориентированного подхода. Бизнес-процесс, представленный в форме структурной или объектной модели, использует алфавит и синтаксис конкретного языка моделирования, что позволяет описывать процесс независимо от предметной области [6].

Следуя обозначениям [1, 2], рассмотрим систему полиномиальных уравнений

$$P_j(z,x)=0, \quad P_j(0,0)=0, \quad j=1,\dots,k, \tag{1}$$

которая решается относительно символов $z=(z_1,\dots,z_n)$ в виде формальных степенных рядов, зависящих от символов $x=(x_1,\dots,x_m)$. Системы такого вида имеют приложения в теории формальных языков, поскольку обобщают важные классы грамматик [3, 4].

Символы $x_1,\dots,x_m$ называются терминальными и образуют словарь языка, а символы $z_1,\dots,z_n$ — нетерминальными, они необходимы для задания грамматических правил. Над всеми символами определена некоммутативная операция конкатенации и коммутативная операция формальной суммы, а также коммутативная операция умножения на числа, и потому можно рассматривать ФСР с числовыми коэффициентами. Мономы от терминальных символов интерпретируются как предложения языка, а каждый ФСР (сумма всех «правильных» мономов), который является решением системы (1), понимается как порождённый грамматикой формальный язык [3, 4].

Поскольку исследовать системы с некоммутативными символами трудно, в [1, 2] предложено рассмотреть коммутативный образ системы (1), который получается в предположении, что все переменные коммутативны; обозначим коммутативный образ ФСР s через $\mathrm{ci}(s)$ [5].

В работе [1] рассмотрен коммутативный образ

$$\mathrm{ci}(P_j(z,x))=0, \quad j=1,\dots,k, \tag{2}$$

системы уравнений (1) и отмечено, что из совместности некоммутативной системы (1) следует совместность коммутативной системы (2), а обратное утверждение неверно. Поэтому вопрос о достаточном условии совместности системы (1) остаётся открытым, частично его решает применение такого инструмента, как якобиан.

Пусть $k=n$; $J(z,x)=\det((\mathrm{ci}(P_i(z,x)))'_{z_j})$ — якобиан системы уравнений (2) по переменным $z_1,\dots,z_n$. Ранее была доказана следующая

Теорема 1 [1]. Если для некоммутативной символьной системы уравнений (1) выполнено неравенство $J(0,0)\neq 0$, то она имеет единственное решение в виде ФСР.

Естественно, она неприменима, когда якобиан $J(0,0)$ равен нулю, однако обойти эту ситуацию в ряде случаев помогают геометрические соображения. Имеет место следующий результат:

Теорема 2. Если для некоммутативной символьной системы уравнений (1) множества $\{\mathrm{ci}(P_j(z,0))=0\}$ в n-мерном комплексном пространстве являются гладкими комплексными аналитическими поверхностями в точке 0, $j=1,\dots,n$, а нормали к ним, проведённые из этой точки, линейно независимы, то система (1) имеет единственное решение в виде ФСР.

Таким образом, теорема 2, обобщая теорему 1, позволяет установить, когда грамматика действительно порождает формальный язык. При этом компонентами решения системы (2) являются алгебраические функции, которые могут быть исследованы аналитически [6 – 8].

ЛИТЕРАТУРА

1. *Егорушкин О. И., Колбасина И. В., Сафонов К. В.* О совместности систем символьных полиномиальных уравнений и их приложении // Прикладная дискретная математика. Приложение. 2016. № 9. С. 119–121.
2. *Egorushkin O. I., Kolbasina I. V., and Safonov K. V.* On solvability of systems of symbolic polynomial equations // Журн. СФУ. Сер. Матем. и физ. 2016. Т. 9. Вып. 2. С. 166–172.
3. *Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л.* Алгебра. Языки. Программирование. Киев: Наукова думка, 1973.
4. *Salomaa A. and Soitolla M.* Automata-Theoretic Aspects of Formal Power Series. N.Y.: Springer Verlag, 1978.
5. *Семёнов А. Л.* Алгоритмические проблемы для степенных рядов и контекстно-свободных грамматик // Доклады АН СССР. 1973. № 212. С. 50–52.
6. *Сафонов К. В., Егорушкин О. И.* О синтаксическом анализе и проблеме В. М. Глушкова распознавания контекстно-свободных языков Хомского // Вестник Томского госуниверситета. 2006. Приложение № 17. С. 63–67.
7. *Сафонов К. В.* Об условиях алгебраичности и рациональности суммы степенного ряда // Матем. заметки. 1987. Т. 41. Вып. 3. С. 325–332.
8. *Safonov K. V.* On power series of algebraic and rational functions in C^n // J. Math. Analysis Appl. 2000. V. 243. P. 261–277.

УДК 519.682 DOI 10.17223/2226308X/13/32

АЛГОРИТМ РЕШЕНИЯ РАСШИРЕННОЙ ПРОБЛЕМЫ СИНТАКСИЧЕСКОГО АНАЛИЗА

В. В. Кишкан, К. В. Сафонов

Уточняется формулировка расширенной проблемы синтаксического анализа: разработать беступиковый алгоритм, который позволяет установить, может ли данный моном быть выведен при помощи системы продукций, образующих грамматику контекстно-свободного языка программирования, а также описать сразу все выводы этого монома, если такие существуют. Описание вывода монома состоит в следующем: определить, какие продукции из грамматики языка, сколько раз и в каком порядке применяются, что равносильно построению всех деревьев вывода. Предложен алгоритм решения расширенной проблемы синтаксического анализа, основанный на иерархии маркированных скобок; маркировка скобок показывает, за какой продукцией они закреплены, и позволяет проследить порядок их использования. Алгоритм имеет простую программную реализацию, дана также оценка сложности алгоритма.

Ключевые слова: *расширенная проблема синтаксического анализа, контекстно-свободный язык, сложность алгоритма.*

При разработке перспективных языков программирования, в том числе предназначенных для обеспечения работы суперкомпьютеров, включая квантовые, возникает необходимость исследовать контекстно-свободные языки (кс-языки) и контекстно-свободные грамматики (кс-грамматики). Один из аспектов связан с проблемой синтаксического анализа (разбора) выражений, написанных на языке программирования. Обычно для синтаксического анализа используются специальные программы — парсеры, разработанные применительно к тому или иному языку программирования и основанные на определённых алгоритмах разбора. Однако в ситуации, когда разра-

батывается и тестируется новый язык программирования, никаких парсеров ещё нет. Когда необходимо провести синтаксический анализ некоторого выражения относительно совокупности грамматических правил, находящихся в стадии разработки, могут быть полезными различные алгоритмы, в том числе имеющие высокую сложность — как правило, анализируются выражения ограниченной длины, и в этом случае высокая сложность алгоритма не играет решающей роли. Если длина тестируемой программы не слишком велика, сложность алгоритма синтаксического разбора может быть даже выше экспоненциальной — важно лишь, чтобы алгоритм был вполне конструктивным и допускал простую программную реализацию.

Отметим, что практически все известные в настоящее время языки программирования являются кс-языками, порождёнными кс-грамматиками, и потому кс-язык является адекватной математической моделью любого языка программирования, в которой правильным программам отведена роль мономов кс-языка.

Проблема синтаксического анализа мономов кс-языка возникла на заре теории языков программирования в 50–60-х годах прошлого века. Удивительно, но до настоящего времени в формулировке проблемы сохраняются разночтения, в связи с чем возникает необходимость уточнить её, а именно: рассмотрим кс-язык, порождённый кс-грамматикой, которая представляет собой систему правил вывода (продукций)

$$z_j \to\ q_{j1}(z,x),\ \ldots,\ z_j \to\ q_{jp_j}(z,x),\ j=1,\ \ldots,n, \tag{1}$$

где $q_{jk}(z,x)$ — мономы от некоммутативных символов алфавита $z_1\ldots,z_n,\ x_1,\ldots,x_m$ с числовым коэффициентом равным 1.

Символы $x_1,\ldots,x_m$ из второй группы называются терминальными символами и образуют словарь кс-языка. Применительно к языкам программирования терминальными символами являются цифры, буквы, вспомогательные знаки, а также состоящие из них «блоки», обозначающие, например, операторы языка программирования. Символы первой группы $z_1,\ldots,z_n$ называются нетерминальными, поскольку не присутствуют явно в тексте программ, а играют вспомогательную роль, участвуя в кс-грамматике как совокупности продукций, порождающих кс-язык.

По правилам грамматики формируются мономы от терминальных символов $x_1,\ldots,x_m$, которые интерпретируются как правильные предложения языка [1, 2]. Такие мономы рассматриваются как корректные, в отличие от произвольных мономов, которые могут не соответствовать правилам грамматики, а значит, являются некорректными.

Вывод корректных мономов кс-языка с помощью системы продукций (1) осуществляется так: продукции сначала применяются к начальному символу z_1, а затем к другим получающимся мономам неограниченное число раз и в любом порядке, что позволяют продуцировать новые мономы от терминальных и нетерминальных символов. Вывод заканчивается, когда получается моном только от терминальных символов — это и есть корректный моном языка, дальнейший вывод из него невозможен, поскольку продукции применимы только к нетерминальным символам. Все корректные мономы образуют соответствующий кс-язык.

В проблеме синтаксического анализа мономов кс-языка выделяют две составляющие: первая часть, называемая проблемой принадлежности или этапом синтаксического контроля, состоит в том, чтобы определить, принадлежит ли данный моном рассматриваемому кс-языку, т. е. может ли быть получен из начального символа z_1 при помощи продукций; вторая часть проблемы — описание синтаксической структуры монома. Такое описание понимается в литературе по-разному.

Так, различные авторы при постановке проблемы синтаксического анализа допускают следующие варианты: требуется разработать алгоритм для того, чтобы установить, какие правила подстановки и сколько раз использовались при выводе данного монома, при этом порядок использования правил подстановки не имеет значения; какие правила подстановки, сколько раз и в каком порядке использовались при выводе этого монома, т. е. построить хотя бы один из возможных выводов монома [2]. Как видно, для полного решения проблемы синтаксического анализа необходимо построить сразу все возможные выводы монома, если таких несколько.

Кроме того, исследователи уделяют большое внимание тому, чтобы разработать беступиковый алгоритм синтаксического анализа мономов кс-языка [1, с. 248]).

В связи с этим будем называть *расширенной проблемой синтаксического анализа* мономов кс-языка проблему разработки беступикового алгоритма, который позволяет установить, может ли моном быть выведен при помощи системы продукций кс-языка (решить проблему принадлежности), а также найти сразу все выводы этого монома; описание вывода монома будем понимать следующим образом: определить, какие продукции, сколько раз и в каком порядке применяются для вывода этого монома, что равносильно построению всех деревьев вывода.

Синтаксический анализ монома, проводимый в соответствии с этим алгоритмом, будем называть *расширенным синтаксическим анализом* (алгоритм 1).

Для решения расширенной проблемы синтаксического анализа рассмотрим расширенную систему уравнений Хомского — Шютценберже, которая имеет вид

$$z_j = Q_j^*(z,x,t) = t_{j1}[\ q_{j1}(z,x)\ ] + \ \cdots\ + t_{jp_j}[\ q_{jp_j}(z,x)\ ],\ j = 1,\ \ldots\ , n.$$

Решение этой системы можно получить методом последовательных приближений [2]:

$$z^{(k+1)}(x,t) = Q^*(z^{(k)}(x,t),x,t);\ k = 0,1,\ldots;\ z^{(0)} = 0. \tag{2}$$

В результате решение получается в виде формальных степенных рядов

$$z_j = z_j^*(x,t) = \sum_{i=0}^{\infty} \langle z_j^*, w_i \rangle\ w_i,\ j = 1,\ldots,n,$$

где w_i — мономы от символов $x_1,\ldots,x_m,t_{11},t_{12},\ldots,t_{np_n}$ с числовыми коэффициентами $\langle z_j^*, w_i\rangle$, содержащие также систему открывающихся и закрывающихся скобок.

Алгоритм 1. Решение расширенной проблемы синтаксического анализа

Вход: моном (программа) w степени (длины) N.

1: Проводим N итераций метода последовательных приближений (2) для решения соответствующей расширенной системы уравнений Хомского — Шютценберже.

2: Перебираем все полученные в шаге 1 мономы степени N, определяя те из них, которые, с точностью до множителей t_{jk}, совпадают с мономом w.
Если таких мономов нет, то моном w вывести невозможно; если есть, то они дают решение расширенной проблемы синтаксического анализа в соответствии со следующим шагом.

3: Считываем все найденные в шаге 2 мономы слева направо, устанавливая иерархию маркированных скобок (по признаку сравнения скобок — внутренняя или внешняя) и определяя тем самым порядок применения продукций при выводе монома w.

Теорема 1. Сложность алгоритма 1 равна $O\left(Ng^{d^N}\right)$, где g и d — некоторые целые числа.

Несмотря на то, что сложность данного алгоритма высокая, он может быть использован для синтаксического анализа применительно к языку программирования, находящемуся в стадии разработки.

ЛИТЕРАТУРА

1. *Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л.* Алгебра. Языки. Программирование. Киев: Наукова думка, 1973.
2. *Salomaa A. and Soitolla M.* Automata-Theoretic Aspects of Formal Power Series. N.Y.: Springer Verlag, 1978.

УДК 510.52 DOI 10.17223/2226308X/13/33

О ГЕНЕРИЧЕСКОЙ СЛОЖНОСТИ ПРОБЛЕМЫ ПРЕДСТАВИМОСТИ НАТУРАЛЬНЫХ ЧИСЕЛ СУММОЙ ДВУХ КВАДРАТОВ

А. Н. Рыбалов

Изучается генерическая сложность проблемы представимости натуральных чисел суммой двух квадратов. Эта проблема, восходящая ещё к Ферма и Эйлеру, тесно связана с проблемами факторизации целых чисел и распознавания квадратичности вычетов по составным модулям, для решения которых не известно эффективных алгоритмов. Доказывается, что, при условии трудноразрешимости этой проблемы в худшем случае и P = BPP, для её решения не существует полиномиального сильно генерического алгоритма. Сильно генерический алгоритм решает проблему не на всём множестве входов, а на подмножестве, последовательность относительных плотностей которого при увеличении размера экспоненциально быстро сходится к 1.

Ключевые слова: *генерическая сложность, суммы квадратов, диофантовы уравнения.*

Введение

Проблема представимости натуральных чисел суммой двух квадратов состоит в том, чтобы по любому заданному натуральному числу N определить, разрешимо ли в натуральных числах диофантово уравнение $x^2 + y^2 = N$. Эта задача восходит ещё к Ферма, который в 1640 г. сформулировал (см. [1, 2]) следующее красивое утверждение: любое простое число вида $p = 4n + 1$ представимо в виде суммы квадратов двух натуральных чисел. Эта гипотеза впоследствие была доказана Эйлером и называется теперь теоремой Ферма — Эйлера [1, 2]. В дальнейшем был получен критерий Ферма — Эйлера разрешимости диофантова уравнения $x^2 + y^2 = N$ для любого натурального N. Однако этот критерий сводит проблему к задаче факторизации (разложения на множители) целых чисел, которая на текщий момент считается трудноразрешимой [3]. Таким образом, критерий Ферма — Эйлера не может быть проверен эффективно (за полиномиальное от размера входа время).

Генерический подход к алгоритмическим проблемам предложен в [4]. В рамках этого подхода алгоритмическая проблема рассматривается не на всём множестве входов, а на некотором подмножестве «почти всех» входов. Такие входы образуют генерическое множество. Понятие «почти все» формализуется введением естественной меры на

множестве входных данных. С точки зрения практики алгоритмы, решающие быстро проблему на генерическом множестве, так же хороши, как и быстрые алгоритмы для всех входов. Классическим примером такого алгоритма является симплекс-метод — он за полиномиальное время решает задачу линейного программирования для большинства входных данных, но имеет экспоненциальную сложность в худшем случае. Более того, может так оказаться, что проблема трудноразрешима или вообще неразрешима в классическом смысле, но легкоразрешима на генерическом множестве.

В данной работе изучается генерическая сложность проблемы представимости натуральных чисел суммой двух квадратов. Доказывается, что если проблема трудноразрешима в худшем случае и P = BPP, то для неё не существует полиномиального сильно генерического алгоритма. Класс BPP состоит из проблем, разрешимых за полиномиальное время на вероятностных машинах Тьюринга. Одной из важных гипотез в теории сложности вычислений является гипотеза о совпадении классов P и BPP. Из неё следует, что любой полиномиальный вероятностный алгоритм $\mathcal{A}$ можно эффективно дерандомизировать, то есть построить полиномиальный алгоритм $\mathcal{B}$, не использующий генератор случайных чисел и решающий ту же проблему, что и алгоритм $\mathcal{A}$. В работе [5] доказано, что равенство P = BPP следует из весьма правдоподобных гипотез о вычислительной сложности некоторых трудных проблем.

1. Генерические алгоритмы

Пусть I — некоторое множество входов. Для подмножества $S \subseteq I$ определим *последовательность относительных плотностей*

$$\rho_n(S) = \frac{|S_n|}{|I_n|},\ n = 1, 2, 3, \ldots,$$

где I_n — множество входов размера n; $S_n = S \cap I_n$. Заметим, что $\rho_n(S)$ — это вероятность попасть в S при случайной и равновероятной генерации входов из I_n. В данной работе множеством входов для алгоритмов является множество натуральных чисел, записанных в двоичной форме. Под размером натурального числа понимается длина его двоичной записи.

Асимптотической плотностью множества S назовём верхний предел

$$\rho(S) = \overline{\lim_{n\to\infty}} \rho_n(S).$$

Множество S называется *генерическим*, если $\rho(S) = 1$, и *пренебрежимым*, если $\rho(S) = 0$. Очевидно, что S генерическое тогда и только тогда, когда его дополнение $I \setminus S$ пренебрежимо.

Следуя [4], назовём множество S *сильно пренебрежимым*, если последовательность $\rho_n(S)$ экспоненциально быстро сходится к 0, т. е. существуют константы σ, $0 < \sigma < 1$, и $C > 0$, такие, что $\rho_n(S) < C\sigma^n$ для любого n. Теперь S называется *сильно генерическим*, если его дополнение $I \setminus S$ сильно пренебрежимо.

Алгоритм $\mathcal{A}$ с множеством входов I и множеством выходов $J \cup \{?\}$ $(? \notin J)$ называется (*сильно*) *генерическим*, если

1) $\mathcal{A}$ останавливается на всех входах из I;
2) множество $\{x \in I : \mathcal{A}(x) \neq ?\}$ является (сильно) генерическим.

Генерический алгоритм $\mathcal{A}$ вычисляет функцию $f : I \to J$, если $(\mathcal{A}(x) = y \in J) \Rightarrow$ $\Rightarrow (f(x) = y)$ для всех $x \in I$. Ситуация $\mathcal{A}(x) = ?$ означает, что $\mathcal{A}$ не может вычислить функцию f на аргументе x. Но условие 2 гарантирует, что $\mathcal{A}$ корректно вычисляет f на

почти всех входах (входах из генерического множества). Множество $S \subseteq I$ называется (*сильно*) *генерически разрешимым за полиномиальное время*, если существует (сильно) генерический полиномиальный алгоритм, вычисляющий его характеристическую функцию.

2. Проблема представимости натуральных чисел суммой двух квадратов

Проблема представимости натуральных чисел суммой двух квадратов состоит в следующем. Дано натуральное число N, записанное в двоичной системе. Нужно определить, разрешимо ли в натуральных числах диофантово уравнение $x^2 + y^2 = N$. Классический критерий Ферма — Эйлера [1, 2] связывает эту проблему с известной проблемой факторизации целых чисел.

Теорема 1 (Ферма, Эйлер). Пусть N — натуральное число. Диофантово уравнение $N = x^2 + y^2$ разрешимо в натуральных числах тогда и только тогда, когда каждый простой делитель N вида $4k + 3$ входит в разложение N в чётной степени.

Если бы проблема факторизации решалась эффективно, то этот критерий давал бы эффективный алгоритм для проблемы представимости натуральных чисел суммой двух квадратов. Однако до сих пор неизвестно эффективных алгоритмов для проблемы факторизации [3]. Кроме того, проблема представимости натуральных чисел суммой двух квадратов тесно связана с проблемой распознавания квадратичности вычетов по составным модулям, которая тоже считается трудноразрешимой [3].

3. Основные результаты

Теорема 2. Если существует сильно генерический полиномиальный алгоритм, решающий проблему представимости натуральных чисел суммой двух квадратов, то существует вероятностный полиномиальный алгоритм, разрешающий эту проблему на всём множестве входов.

Теорема 3. Если проблема представимости натуральных чисел суммой двух квадратов не лежит в классе P и P = BPP, то не существует сильно генерического полиномиального алгоритма для этой проблемы.

ЛИТЕРАТУРА

1. *Dickson L. E.* History of the Theory of Numbers. V. II. N.Y.: Dover Publications, 2005. 803 p.
2. *Сендеров В., Спивак А.* Суммы квадратов и целые гауссовы числа // Квант. 1999. № 3. С. 14–22.
3. *Adleman L. M. and McCurley K. S.* Open problems in number theoretic complexity, II // Proc. First Intern. Symp. Algorithmic Number Theory. N.Y., USA, May 6–9, 1994. P. 291–322.
4. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks // J. Algebra. 2003. V. 264. No. 2. P. 665–694.
5. *Impagliazzo R. and Wigderson A.* P = BPP unless E has subexponential circuits: Derandomizing the XOR Lemma. Proc. 29th STOC. El Paso: ACM, 1997. P. 220–229.

Секция 7

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 519.7 DOI 10.17223/2226308X/13/34

ПРИМЕНЕНИЕ SAT-ОРАКУЛОВ ДЛЯ ГЕНЕРАЦИИ ДОПОЛНИТЕЛЬНЫХ ЛИНЕЙНЫХ ОГРАНИЧЕНИЙ В ЗАДАЧАХ КРИПТОАНАЛИЗА НЕКОТОРЫХ ЛЕГКОВЕСНЫХ ШИФРОВ[1]

К. В. Антонов, А. А. Семёнов

Описывается новая техника, применимая к задачам алгебраического криптоанализа. В рамках предлагаемой техники строятся линейные уравнения над полем из двух элементов, которыми дополняется система алгебраических уравнений, представляющая криптоанализ рассматриваемого шифра. Для генерации новых линейных уравнений используется SAT-решатель. Показано, что применение этой техники позволяет повысить эффективность атак из класса «угадывай и определяй», основанных на понятии линеаризующего множества. Эффективность предложенной техники подтверждается вычислительными экспериментами, проведёнными для ряда ослабленных по числу шагов инициализации версий известного поточного шифра Trivium.

Ключевые слова: *линеаризующие множества, атаки из класса «угадывай и определяй», квадратичные системы над GF(2), псевдобулева оптимизация, Trivium.*

Настоящую работу можно рассматривать как прямое продолжение [1]. Приведём краткое описание используемых обозначений, понятий и вспомогательных результатов из [1]. Будем рассматривать задачу обращения (поиска прообразов) всюду определённой дискретной функции

$$f : \{0,1\}^n \to \{0,1\}^m, \tag{1}$$

которая задана некоторой программой (алгоритмом) M_f. Иными словами, зная текст программы M_f и произвольный $\gamma \in \operatorname{Range} f$, требуется найти такой $\alpha \in \{0,1\}^n$, что $f(\alpha) = \gamma$. Известно, что по M_f можно эффективно построить схему из функциональных элементов над базисом $\{\wedge, \neg\}$ (конъюнкция, отрицание), которая задаёт функцию f. Такого рода схемы в символьной верификации называются И-НЕ-графами (And-Inverter Graph или AIG [2]). На практике для построения И-НЕ-графа по конкретному алгоритму M_f можно задействовать специализированные программные средства. Мы использовали для этих целей программный комплекс Transalg [3, 4].

Пусть G_f — И-НЕ-граф, который задаёт функцию (1). В G_f выделены n вершин, не имеющие предшественников (соответствуют аргументу функции f), эти вершины называются входными. Всем остальным вершинам G_f приписаны функциональные элементы из базиса $\{\wedge, \neg\}$, эти вершины называются внутренними вершинами или узлами. Среди внутренних вершин выделены m вершин, не имеющих потомков, эти вер-

[1]Работа выполнена при финансовой поддержке Российского научного фонда, проект № 16-11-10046.

шины соответствуют значению функции f и называются выходными. С входными вершинами G_f связываются булевы переменные, образующие множество $X = \{x_1, \ldots, x_n\}$. С каждой внутренней вершиной g графа G_f связывается булева переменная v, называемая вспомогательной, множество всех вспомогательных переменных обозначается через V. В множестве V выделяется подмножество $Y = \{y_1, \ldots, y_m\}$, образованное переменными, которые приписаны выходным вершинам G_f.

По графу G_f, используя преобразования Цейтина [5], можно построить КНФ C_f, которая называется шаблонной (template CNF [6]). Также по графу G_f можно построить систему алгебраических уравнений над GF(2), каждое уравнение которой имеет степень не выше 2. Опишем кратко соответствующую процедуру. Рассматриваем множества переменных X и V. Каждой вершине g с приписанной ей переменной $v \in V$ сопоставим алгебраическое уравнение над полем GF(2). Если g — И-узел, то g имеет двух прямых предшественников в графе G_f. Предположим, что им приписаны переменные u и w. Если g — НЕ-узел, то он имеет в G_f единственного прямого предшественника, которому приписана переменная u. Произвольному g сопоставим уравнение над GF(2) по следующим правилам. Если g — И-узел, то имеем уравнение

$$u \wedge w \oplus v = 0, \tag{2}$$

если g — НЕ-узел, то имеем уравнение

$$u \oplus v = 1. \tag{3}$$

Определение 1. Пусть E_f — система, образованная уравнениями вида (2) или (3) по всем узлам графа G_f. Назовём E_f шаблонной системой уравнений над GF(2) для функции (1).

Заметим, что E_f образована уравнениями над GF(2) степени не выше 2. Стандартным образом [7, 8] определим для произвольной переменной $x \in U$, $U = X \cup V$, подстановку её значения $x = \lambda \in \{0, 1\}$ в систему E_f. Иногда в результате подстановки $x = \lambda$ вид некоторого уравнения может упроститься таким образом, что станет известным значение некоторой переменной $x' \in X \setminus \{x\}$. В таких случаях будем говорить, что соответствующее значение переменной x' индуцировано подстановкой $x = \lambda$. Например, подстановка $u = 1$ в (3) индуцирует значение 0 переменной v.

Аналогичным образом определяется произвольная подстановка вида $x = \lambda$ в шаблонную КНФ C_f. В [6] показано, что подстановка в C_f набора γ значений переменных из Y, $\gamma \in \operatorname{Range} f$, даёт выполнимую КНФ $C_f(\gamma)$, из выполняющего набора которой эффективно извлекается такое $\alpha \in \{0, 1\}^n$, что $f(\alpha) = \gamma$. Рассуждая по аналогии, можно показать, что подстановка $\gamma \in \operatorname{Range} f$ в E_f даёт совместную систему уравнений $E_f(\gamma)$ над GF(2) и из произвольного решения $E_f(\gamma)$ можно эффективно извлечь такое $\alpha \in \{0, 1\}^n$, что $f(\alpha) = \gamma$.

Понятие линеаризующего множества сформулировано в [1]. Оно обобщает понятие линеаризационного множества, введённого в [8]. Неформально говоря, линеаризующее множество линеаризует систему вида $E_f(\gamma)$ с некоторой вероятностью, которая может быть существенно меньше 1, но давать при этом атаку с относительно малой трудоёмкостью. Более точно, линеаризующее множество определяется на базе конструкции, с использованием которой в [9] предложены новые атаки из класса «угадывай и определяй». В соответствии с этой конструкцией с произвольным α, которое выбирается из $\{0, 1\}^n$ согласно равномерному распределению, и произвольным $B \subseteq X$ связывается

набор значений переменных β_α (получается в результате выбора соответствующих B компонент из α), а также $\gamma_\alpha \in \operatorname{Range} f$, такой что $f(\alpha) = \gamma_\alpha$. Вероятность линеаризации p_B — это доля таких $\alpha \in \{0,1\}^n$, что подстановка пары β_α, γ_α в систему E_f превращает её в линейную.

В [1] задача поиска линеаризующего множества с относительно малой трудоёмкостью соответствующей атаки ставится как проблема оптимизации специальной псевдобулевой функции [10], значения которой вычисляются в результате вероятностного эксперимента. В [1] для этой цели используется алгоритм, основанный на концепции tabu search [11], а в [12] — один вариант генетического алгоритма, описанный в [13]. Как итог, для задачи криптоанализа генератора A5/1 найдены линеаризующие множества, дающие атаки, трудоёмкость которых существенно меньше трудоёмкости известной атаки Р. Андерсона.

В настоящей работе описана техника, которая позволяет дополнять системы вида $E_f(\gamma)$ новыми линейными уравнениями, что в ряде случаев позволяет построить существенно более эффективные (в смысле трудоёмкости соответствующих атак) линеаризующие множества.

Итак, рассматривается задача обращения функции вида (1). Предположим, что функция f представлена в виде И-НЕ-графа G_f и по G_f построены шаблонная КНФ C_f и шаблонная система уравнений E_f над GF(2). Таким образом, и в C_f, и в E_f фигурируют переменные, образующие множество $U = X \cup V$.

Рассмотрим произвольный И-узел g в графе G_f. Пусть узлу g приписана переменная v, а прямым предшественникам g — переменные u и w. С узлом g связана булева функция $\varphi_g : \{0,1\}^3 \to \{0,1\}$, заданная формулой $u \wedge w \equiv v$. При построении C_f формула $u \wedge w \equiv v$ приводится к КНФ по таблице T_g (табл. 1).

Таблица 1

Табличное задание функции φ_g

u	w	v	φ_g
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Обозначим через S_g множество решений связанного с g уравнения $u \wedge w \oplus v = 0$. Очевидно, что S_g образовано всеми теми наборами значений переменных u, w, v, которым в таблице T_g соответствует $\varphi_g = 1$. С другой стороны, в соответствии с преобразованиями Цейтина при построении C_f в эту КНФ войдут дизъюнкции вида $u^{\neg\sigma_1} \vee w^{\neg\sigma_2} \vee v^{\neg\sigma_3}$ по всем таким наборам $(\sigma_1\sigma_2\sigma_3)$ из T_g, на которых $\varphi_g = 0$.

В основе приведённого далее результата лежит следующее наблюдение: оказывается, для целого ряда криптографических функций вида (1) для существенной доли И-узлов g в G_f в множестве S_g существуют такие наборы $(\sigma_1\sigma_2\sigma_3)$, что КНФ $u^{\sigma_1} \wedge w^{\sigma_2} \wedge v^{\sigma_3} \wedge C_f$ невыполнима. С использованием рассуждений из [6] можно показать, что данная ситуация соответствует тому факту, что никакой вход $\alpha \in \{0,1\}^n$ не может индуцировать для переменных из множества $\{u, w, v\}$ значение $(\sigma_1\sigma_2\sigma_3)$. Применительно к системе вида $E_f(\gamma)$ для произвольного $\gamma \in \operatorname{Range} f$ это означает,

что вектор $(\sigma_1\sigma_2\sigma_3)$ может быть заведомо исключён из возможных решений данной системы. Может показаться удивительным, но, как правило, на доказательство невыполнимости КНФ вида $u^{\sigma_1} \wedge w^{\sigma_2} \wedge v^{\sigma_3} \wedge C_f$ у современного SAT-решателя уходят доли секунды. Таким образом, можно говорить, что такой SAT-решатель выполняет роль оракула, эффективно отсеивающего некоторые наборы из S_g. Основной результат настоящей работы состоит в следующем.

Теорема 1. Пусть g — И-узел в И-НЕ-графе G_f, представляющем произвольную функцию f вида (1); $X_g = \{u, w, v\}$ — множество переменных, связанных с g; S_g — множество решений уравнения $u \wedge w \oplus v = 0$. Предположим, что для некоторого $\sigma = (\sigma_1\sigma_2\sigma_3)$, $\sigma \in S_g$, SAT-оракул доказал невыполнимость КНФ $u^{\sigma_1} \wedge w^{\sigma_2} \wedge v^{\sigma_3} \wedge C_f$. Тогда для любого такого σ имеет место

$$S_g \setminus \{\sigma\} = S_g \cap S_{L(X_g)},$$

где $S_{L(X_g)}$ — множество решений некоторого линейного уравнения $L(X_g)$ над F(2).

Доказательство данной теоремы получается в результате разбора всех возможных случаев исключения σ из S_g. Множество S_g приведено в табл. 2.

Таблица 2

Множество S_g для И-узла g

u	w	v	φ_g
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Важное следствие теоремы 1 состоит в том, что если для И-узла g графа G_f SAT-оракул доказал невыполнимость КНФ $u^{\sigma_1} \wedge w^{\sigma_2} \wedge v^{\sigma_3} \wedge C_f$ для некоторого набора $\sigma = (\sigma_1\sigma_2\sigma_3)$, $\sigma \in S_g$, то для любого $\gamma \in \mathrm{Range}\, f$ можно добавить к системе $E_f(\gamma)$ некоторое линейное уравнение и получить эквивалентную систему. Тем самым использование шаблонной КНФ C_f и SAT-оракула позволяет эффективно добавлять в системы вида $E_f(\gamma)$ новые линейные уравнения.

Алгоритм, проверяющий для каждого И-узла g в G_f выполнимость КНФ $u^{\sigma_1} \wedge \wedge w^{\sigma_2} \wedge v^{\sigma_3} \wedge C_f$ по всем $\sigma = (\sigma_1\sigma_2\sigma_3)$, $\sigma \in S_g$, реализован в виде программы на C++. В вычислительных экспериментах описанная техника тестировалась на задачах криптоанализа ослабленных по числу шагов инициализации вариантов известного шифра Trivium [14]. Отметим, что Trivium является одним из победителей конкурса eSTREAM, и для данного шифра не известно на сегодняшний день убедительных атак, позволяющих найти секретный ключ существенно быстрее, чем полным перебором. Особенность Trivium заключается в том, что перед генерацией ключевого потока в этом шифре выполняется стадия инициализации, в ходе которой секретный ключ длиной 80 бит смешивается с несекретной 80-битной инициализирующей последовательностью. Изначально в стандарте Trivium предусмотрено 1152 шага инициализации. Однако даже при существенно меньшем числе шагов инициализации получаемые варианты Trivium оказываются стойкими ко всем известным видам криптоанализа. По-видимому, лучшими известными атаками на ослабленные по числу шагов инициализации варианты Trivium являются т. н. «кубические атаки», описанные в [15]. Следует отметить, что атаки из [15] весьма специфичны по ряду моментов. В частности,

предполагается, что противник ищет ключ, который использовался многократно совместно с различными инициализирующими векторами. В атаках, построенных нами, мы исходим из более реалистичного сценария — предполагается, что различные ключи могут использоваться совместно с некоторым фиксированным инициализирующим вектором.

Более конкретно, рассмотрены варианты Trivium с числом шагов инициализации $N = 160, 192, 288, 384$. Для каждого случая решается задача обращения функции

$$f_{(\mathrm{Tr},N)} : \{0,1\}^{80} \to \{0,1\}^{300},$$

которая соответствует алгоритму Trivium с числом шагов инициализации N и известным инициализирующим вектором (во всех экспериментах использовался один и тот же инициализирующий вектор).

Для каждого из полученных шифров мы рассматривали задачу поиска линеаризующего множества с минимальной трудоёмкостью в двух вариантах. В первом варианте использован подход [1, 12]: мы искали множество, линеаризующее систему квадратичных уравнений над GF(2), построенную по И-НЕ-графу $G_{f_{(\mathrm{Tr},N)}}$. Во втором варианте к такой системе добавляются дополнительные линейные уравнения, сгенерированные при помощи SAT-оракула в соответствии с описанной выше техникой.

Задача поиска эффективного линеаризующего множества ставится как задача минимизации псевдобулевой функции, описанной в [12]. Для её решения используется генетический алгоритм [13]. Вычислительные эксперименты проводились на кластере «Академик В. М. Матросов» Иркутского суперкомпьютерного центра СО РАН [16]. Результаты экспериментов в виде оценок трудоёмкости соответствующих атак приведены в табл. 3.

Таблица 3

Результаты сравнения двух подходов

N	Метод	$\lvert B\rvert$	p_B	Сложность атаки (число решённых систем уравнений)
160	Алгоритм из [12]	44	0,774	6,82e+13
	Алгоритм с SAT-оракулом	39	0,350	4,71e+12
192	Алгоритм из [12]	58	0,431	2,01e+18
	Алгоритм с SAT-оракулом	48	0,219	3,86e+15
288	Алгоритм из [12]	73	0,506	5,60e+22
	Алгоритм с SAT-оракулом	66	0,457	4,84e+20
384	Алгоритм из [12]	78	0,954	9,50e+23
	Алгоритм с SAT-оракулом	74	0,093	6,12e+23

Комментарии к табл. 3. В первом столбце приведено число шагов инициализации в рассматриваемой версии шифра Trivium. Во втором столбце указаны алгоритмы: мы сравниваем метод, описанный в [12], с методом, представленным в настоящей работе (с использованием SAT-оракула). В последующих столбцах приведены мощность линеаризующего множества, вероятность линеаризации и оценка числа систем линейных уравнений, которые необходимо решить для нахождения 80-битного секретного ключа.

ЛИТЕРАТУРА

1. *Семёнов А. А., Антонов К. В., Отпущенников И. В.* Поиск линеаризующих множеств в алгебраическом криптоанализе как задача псевдобулевой оптимизации // Прикладная дискретная математика. Приложение. 2019. № 12. С. 130–134.

2. *Biere A.* Bounded Model Checking // Handbook of Satisfiability. Amsterdam: IOS Press, 2009. P. 457–481.
3. *Отпущенников И. В., Семёнов А. А.* Технология трансляции комбинаторных проблем в булевы уравнения // Прикладная дискретная математика. 2011. № 1 (11). С. 96–115.
4. *Otpuschennikov I., Semenov A., Gribanova I., et al.* Encoding cryptographic functions to SAT using TRANSALG system // Proc. 22nd European Conf. ECAI 2016. Frontiers in Artificial Intelligence and Applications. 2016. V. 285. P. 1594–1595.
5. *Цейтин Г. С.* О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ АН СССР. 1968. Т. 8. С. 234–259.
6. *Semenov A., Otpuschennikov I., Gribanova I., et al.* Translation of algorithmic descriptions of discrete functions to SAT with application to cryptanalysis problems // Log. Methods Comput. Sci. 2020. V. 16. Iss. 1. P. 29:1–29:42.
7. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983.
8. *Агибалов Г. П.* Логические уравнения в криптоанализе генераторов ключевого потока // Вестник Томского госуниверситета. Приложение. 2003. № 6. С. 31–41.
9. *Semenov A., Zaikin O., Otpuschennikov I., et al.* On cryptographic attacks using backdoors for SAT // Proc. 32nd AAAI Conf. 2018. P. 6641–6648.
10. *Boros E. and Hammer P.* Pseudo-Boolean optimization // Discr. Appl. Math. 2002. V. 123. Iss. 1–3. P. 155–225.
11. *Glover F. and Laguna M.* Tabu Search. Norwell: Kluwer Academic Publishers, 1997.
12. *Антонов К. В., Семёнов А. А.* Применение метаэвристических алгоритмов псевдобулевой оптимизации к поиску линеаризующих множеств в криптоанализе криптографических генераторов // Материалы 6-й Междунар. школы-семинара «Синтаксис и семантика логических систем». Иркутск: ИГУ, 2019. С. 13–18.
13. *Pavlenko A., Semenov A., and Ulyantsev V.* Evolutionary computation techniques for constructing SAT-based attacks in algebraic cryptanalysis // LNCS. 2019. V. 11454. P. 237–253.
14. *De Canniere C.* Trivium: A stream cipher construction inspired by block cipher design principles // LNCS. 2006. V. 4176. P. 171–186.
15. *Dinur I. and Shamir A.* Cube attacks on tweakable black box polynomials // LNCS. 2009. V. 5479. P. 278–299.
16. ЦКП Иркутский суперкомпьютерный центр СО РАН. `http://hpc.icc.ru`.

УДК 519.7 DOI 10.17223/2226308X/13/35

О ДИФФЕРЕНЦИАЛАХ ДЛЯ МОДИФИКАЦИИ ШИФРА SIMON НА ОСНОВЕ СХЕМЫ ЛАЯ — МЕССИ[1]

А. А. Белоусова, Н. Н. Токарева

Рассматриваются блочный итеративный шифр Simon 32/64, основанный на сети Фейстеля, и его модификации на основе схемы Лая — Месси. Получены оценки вероятностей дифференциалов 12 раундов исходного шифра и его модификаций.

Ключевые слова: *схема Лая — Месси, сеть Фейстеля, дифференциальный криптоанализ.*

[1]Работа выполнена в рамках государственного задания Института математики им. С. Л. Соболева СО РАН (проект № 0314-2019-0017) при поддержке РФФИ (проект № 18-07-01394) и Лаборатории криптографии JetBrains Research.

В работе рассматриваются блочные итеративные шифры, основанные на сети Фейстеля (рис. 1) и на альтернативной схеме — схеме Лая — Мэсси [1] (рис. 2). Для исследования выбран шифр Simon 32/64 [2], основанный на сети Фейстеля, и построены две его модификации подстановкой схемы Лая — Мэсси на место сети Фейстеля. Получены оценки вероятностей дифференциалов, построенных для 12 раундов исходной и модифицированных версий шифра Simon 32/64. Оценка вероятности дифференциалов для шифра Simon 32/64 взята из работы [3]: максимальная вероятность дифференциала после прохождения 12 раундов составляет 2^{-36}.

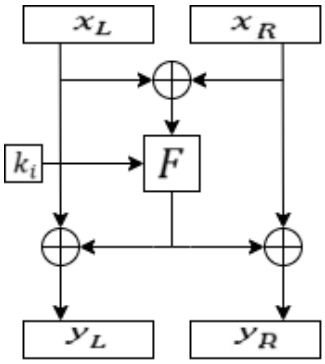

Рис. 1. Сеть Фейстеля

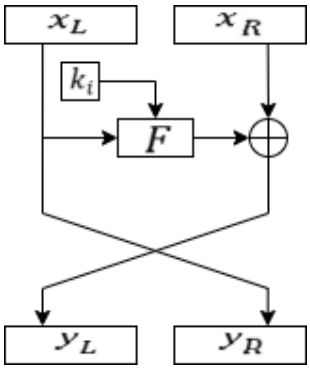

Рис. 2. Схема Лая — Месси

Один раунд схемы Лая — Мэсси в её оригинальном виде записывается как $(y_L, y_R) = (x_L \oplus F(x_L \oplus x_R), x_R \oplus F(x_L \oplus x_R))$, и в этом есть существенный недостаток: для любого входа (x_L, x_R) выполняется соотношение $x_L \oplus x_R = y_L \oplus y_R$, где (y_L, y_R) — выход раунда. В работе [4] отмечено, что для устранения этого недостатка к схеме необходимо добавить перестановку-ортоморфизм σ.

Пусть $\sigma\colon \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ — перестановка на $\mathbb{Z}_2^n$; σ называется *ортоморфизмом* $\mathbb{Z}_2^n$, если $\sigma \oplus I$ — также перестановка на $\mathbb{Z}_2^n$, где I — тождественная перестановка. Тогда один раунд схемы записывается как $(y_L, y_R) = (\sigma(x_L \oplus F(x_L \oplus x_R)), x_R \oplus F(x_L \oplus x_R))$, а разница текстов $y_L \oplus y_R = \sigma(x_L \oplus F(x_L \oplus x_R)) \oplus (x_R \oplus F(x_L \oplus x_R))$.

Проведено сравнение оценок вероятностей дифференциалов [5] оригинальной схемы Лая — Месси и схемы с добавлением ортоморфизма. Для этого написана программа, которая реализует перебор всех разностей открытых текстов. На каждой итерации шифра находится один из наиболее вероятных выходов на раунде с помощью построения строки таблицы дифференциалов, соответствующей входной разности. Далее найденные вероятности перемножаются для получения оценки максимальной вероятности дифференциалов.

После 12 раундов оценка для максимальной вероятности дифференциала для модернизированного шифра Simon32/64 без добавления ортоморфизма составляет 2^{-24}, а с добавлением ортоморфизма находится в интервале между 2^{-24} и 2^{-63}.

Таким образом, оценка максимальной вероятности дифференциала модернизации шифра Simon 32/64 без добавления ортоморфизма выше, чем у оригинального шифра. Компьютерные вычисления на части данных позволяют предположить, что модернизация с ортоморфизмом может быть более устойчивой, чем оригинальный шифр и модернизация без ортоморфизма.

ЛИТЕРАТУРА

1. *Nakahara J.* Lai — Massey Cipher Designs. History, Design Criteria and Cryptanalysis. Springer Nature Switzerland AG, 2018.

2. *Beaulieu R., Shors D., Smith J., et al.* The Simon and Speck Families Of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, 2013.
3. *Abed F., List E., Lucks S., and Wenzel J.* Differential and Linear Cryptanalysis of Reduced-Round Simon. ePrint Archive, Report 2013/526, 2013.
4. *Vaudenay S.* On the Lai — Massey Scheme // ASIACRYPT'99. LNCS. 1999. V. 1716. P. 8–19.
5. *Biham E. and Shamir A.* Differential Cryptanalysis of the Data Encryption Standard. Berlin; Heidelberg: Springer, 1993.

УДК 004.056.55 DOI 10.17223/2226308X/13/36

КОДИРОВАНИЕ ИНФОРМАЦИИ МАТРИЦАМИ УОЛША

М. С. Беспалов, К. М. Малкова

Рассмотрено представление общей линейной группы $\mathrm{GL}(n,2)$ подгруппой автоморфизмов $\mathrm{GL}(N,2)$ при мультипликативной нотации в её действии в пространстве $\mathbb{R}^N$, где $N=2^n$. Каждая матрица как элемент группы $\mathrm{GL}(n,2)$ определяет упорядочения группы $\mathbb{Z}_2^n$ и её группы характеров, популярных при цифровой обработке информации в виде дискретных функций Уолша. На основе быстрого преобразования Уолша и данного соответствия создан программный прототип автоматической системы кодирования выходного сигнала в виде перестановки набора спектральных характеристик.

Ключевые слова: *дискретные функции Уолша, кодовая матрица, быстрое преобразование Уолша, кронекерово произведение.*

Для $\mathbb{Z}_2$ — аддитивной группы поля $\mathbb{F}_2$ — существуют разные представления, среди которых нас интересуют её мультипликативное представление $(\{1,-1\};\cdot)$ и векторное представление $(\{S,A\};\circ)$ с операцией умножения по Адамару векторов $S=(1\ 1)$, $A=(1\ -1)\in\mathbb{R}^2$. Для декартова произведения $\mathbb{Z}_2^n$ группы аддитивное представление рассматривается относительно операции $\oplus$ покоординатного сложения по модулю 2, а мультипликативное — относительно той же операции $\circ$ покоординатного умножения.

Популярные при цифровой обработке сигналов *дискретные функции Уолша* [1, 2] уровня n в работе [3] определены без привлечения нумерации как кронекерово произведение векторов S и A в количестве n сомножителей.

Теорема 1. Множество дискретных функций Уолша уровня n составляет подгруппу G мультипликативной группы $\mathbb{Z}_2^N$, где $N=2^n$, изоморфную группе $\mathbb{Z}_2^n$.

При декартовом произведении $\mathbb{Z}_2^n$ векторного представления $\mathbb{Z}_2=(\{S,A\};\circ)$ элементы будем записывать через разделительный знак $\otimes$, совпадающий с символом кронекерова произведения, что доказывает изоморфизм. Если перейдём к числам $S=(1\ 1)$ и $A=(1\ -1)$ и выполним кронекерово произведение, то получим элементы мультипликативной группы $\mathbb{Z}_2^N$. На основе свойства

$$(u\otimes v)\circ(w\otimes t)=(u\circ w)\otimes(v\circ t),$$

верного для $u,w\in\mathbb{R}^k$, $v,t\in\mathbb{R}^m$, устанавливается их групповое свойство.

Рассмотренная подгруппа $G\subseteq\mathbb{Z}_2^N$ составляет группу характеров конечной абелевой группы $\mathbb{Z}_2^n$, изоморфизм которых вытекает из теории двойственности Понтрягина [4].

Для решения задач цифровой обработки информации [2] эти группы нас интересуют в виде упорядоченных групп. В [5] подробно разбираются три известные нумерации

(Адамара, Пэли и Уолша) дискретных преобразований Уолша и высказывается сожаление об отсутствии других изученных «с точки зрения быстроты сходимости спектров при разложении сигналов и удобств практического применения». Рассматриваются перестановки на $N = 2^n$ элементах в виде двоичной инверсии и кода Грея, организующие переходы между нумерациями.

Определение 1 [3]. Назовём W-матрицей уровня n такую, что все её строки суть различные дискретные функции Уолша уровня n; все её столбцы — различные дискретные функции Уолша уровня n.

Таким образом, любая W-матрица V задаёт два линейных порядка на множестве всех дискретных функций Уолша выбранного уровня: по строкам и по столбцам для двух возможных способов кодирования $y = Vx$ и $y = xV$. Невырожденная матрица K как элемент общей линейной группы $\mathrm{GL}(n, 2)$ также задаёт два аналогичных линейных порядка элементов группы $\mathbb{Z}_2^n$. Так как матрица K, как показано далее, определяет порядок следования отсчётов выходного сигнала при цифровой обработке информации, то назовём её *кодовой матрицей* для данного набора спектральных характеристик.

Теорема 2 [3]. Множество невырожденных булевых матриц порядка n изоморфно множеству W-матриц уровня n.

Указана следующая процедура вычисления W-матрицы по кодовой. Зададим матрицы C_n размера $n \times 2^n$ рекуррентным соотношением

$$C_1 = (0\,1), \quad C_n = \begin{pmatrix} C_{n-1} & C_{n-1} \\ 0_{n-1} & 1_{n-1} \end{pmatrix}, \tag{1}$$

где блоки $0_{n-1} = (0\,0\ldots0)$, $1_{n-1} = (1\,1\ldots1)$ суть строки длины 2^{n-1}. В столбцах матрицы C_n вида (1) лексикографически упорядочены инверсии двоичных кодов чисел от 0 до $2^n - 1$. По формуле (над полем $\mathbb{F}_2$)

$$C_n^{\mathrm{T}} \cdot K \cdot C_n \tag{2}$$

вычислим булеву матрицу порядка $N = 2^n$, в которой произведём перекодировку элементов: $1 \Rightarrow -1$, $0 \Rightarrow 1$.

Обратная процедура выделения кодовой матрицы из W-матрицы: выборкой $(2^0, 2^1, \ldots, 2^{n-1})$ выделим главную подматрицу, в которой произведём обратную перекодировку элементов: $1 \Rightarrow 0$, $-1 \Rightarrow 1$.

Для сокращения записи кодовую матрицу заменяем на *кодовую метку* с записью строк в шестнадцатиричной системе счисления. Все шесть W-матриц уровня 2 явно записаны в [5, 6]. Четыре из них симметричные и соответствуют нумерациям Адамара, Пэли, Уолша и предложенной в [7]. Их кодовые метки 21, 12, 13 и 31 соответственно. Для уровня три их кодовые метки 421, 124, 136 и 652, а для уровня четыре — 8421, 1248, 136C и CA52 соответственно. Общее число W-матриц уровня n, совпадающее с порядком группы $\mathrm{GL}(n, 2)$, вычисляется по формуле $(2^n - 2^0)(2^n - 2^1)\ldots(2^n - 2^{n-1})$.

Известно, что j-я строка произведения матриц равна линейной комбинации строк второго сомножителя с коэффициентами из j-й строки первого сомножителя. По этому правилу реализация левого умножения в (2) организует упорядочение всех элементов векторного пространства $\mathbb{Z}_2^n$ относительно упорядоченного базиса, указанного в строках матрицы K. Правое умножение в (2) организует упорядочение базиса векторного подпространства $G \subset \mathbb{Z}_2^N$. В терминах блочного кодирования [8] произведение

$K \cdot C$ составляет порождающую матрицу для блочного линейного $(2^n, n)$-кода, который (в результате перекодировки $1 \Rightarrow -1$, $0 \Rightarrow 1$) превращается в упорядоченный ортогональный базис пространства $\mathbb{R}^N$ из дискретных функций Уолша уровня n.

Если умножения в формуле (2) рассматривать справа налево, а не стандартно слева направо, то получим аналогичные взаимосвязи для столбцов, а не строк. Известно, что j-й столбец произведения матриц равен линейной комбинации столбцов первого сомножителя с коэффициентами из j-го столбца второго сомножителя. Тогда правое умножение в (2) организует упорядочение всех элементов векторного пространства $\mathbb{Z}_2^n$ в порядке, заданном в столбцах K. Левое умножение в (2) организут упорядочение базиса векторного подпространства $G \subset \mathbb{Z}_2^N$ так, что произведение $C^{\mathrm{T}} \cdot K$ составляет транспонированную порождающую матрицу $(2^n, n)$-кода, переходящую в упорядоченный ортогональный базис пространства $\mathbb{R}^N$.

Авторами создан программный прототип (C#), который моделирует процессы кодирования и декодирования числовых данных с использованием кодовой матрицы. Тип исходных числовых данных, в поле которого будут происходить все программные расчёты, выбирает пользователь, что даёт возможность выделять минимальное необходимое количество байт памяти. В программе предусмотрено задание кодовой метки вручную и случайно. Перед кодированием происходит формирование кодовой матрицы из кодовой метки, её визуализация и проверка на невырожденность. Если матрица невырожденная, то над исходным числовым массивом совершается *быстрое преобразование Уолша* и перестановка элементов выходного массива в порядке, указанном формулой (2) в кодовой матрице. Для декодирования сообщения сначала происходит обратная перестановка элементов, затем *обратное быстрое преобразование Уолша.*

Например, для некоторого сообщения в виде даты выходной сигнал можно выдать в нумерации Пэли $y = (24, 7, 2, 1, 3, 3, 9, 18)$ для кодовой метки 124 или в переставленном виде $y = (24, 3, 9, 2, 1, 18, 3, 7)$ для кодовой метки 463. Предложим разные варианты кодирования. Так как сумма цифр даты меньше 64, для начального отсчёта отведём шесть бит. Так как для представления начального отсчета (числа 24) достаточно пяти, то на каждый из остальных отсчётов алгоритм отводит по пять бит. Получим в первом случае на выходе

$$011000001110001000001000110001101001100 10.$$

Для выходного сигнала с кодовой меткой 463 каждый отсчёт представим в фибоначчиевой системе счисления и получим

$$1000100011000110001011001101101000011000 1101001.$$

На базе дискретных функций Уолша традиционно обрабатывается видео- и аудиоинформация. С помощью системы кодовых меток можно организовать многоканальную систему перенастраивающихся декодеров при передаче скрытой информации по открытым каналам связи.

Характерной особенностью выходного слова служит его представление в алфавите из двух символов (а не трёх) за счёт отсутствия разделительных знаков между отсчётами, которые в данных примерах после простых манипуляций расставляются. Эта особенность важна для массивов с неизвестной заранее разрядностью отсчётов исходного сообщения.

Описанная конструкция допускает p-ичное обобщение, теория которого в виде аналогов приведённых определений, теорем и выносных формул разработана в [9].

ЛИТЕРАТУРА

1. *Малоземов В. Н., Машарский С. М.* Основы дискретного гармонического анализа. СПб.: Лань, 2012.
2. *Залманзон Л. А.* Преобразование Фурье, Уолша, Хаара и их применение в управлении, связи и других областях. М.: Наука, 1989.
3. *Беспалов М. С.* Собственные подпространства дискретного преобразования Уолша // Проблемы передачи информации. 2010. Т. 46. № 3. С. 60–79.
4. *Моррис С.* Двойственность Понтрягина и строение локально компактных абелевых групп. М.: Мир, 1980.
5. *Трахман А. М., Трахман В. А.* Основы теории дискретных сигналов на конечных интервалах. М.: Сов. радио, 1975.
6. *Беспалов М. С., Скляренко В. А.* Дискретные функции Уолша и их приложения. Владимир: ВлГУ. 2014.
7. *Беспалов М. С.* Новая нумерация матриц Уолша // Проблемы передачи информации. 2009. Т. 45. № 4. С. 43–53.
8. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки. М.: Мир, 1976.
9. *Беспалов М. С.* Дискретное преобразование Крестенсона // Проблемы передачи информации. 2010. Т. 46. № 4. С. 91–115.

УДК 519.7 DOI 10.17223/2226308X/13/37

ПРИМЕНЕНИЕ ИНВЕРСНЫХ ЛАЗЕЕК ДЛЯ ПОСТРОЕНИЯ АТАК ИЗ КЛАССА «УГАДЫВАЙ И ОПРЕДЕЛЯЙ» НА ХЕШ-ФУНКЦИИ СЕМЕЙСТВА MD4[1]

И. А. Грибанова, А. А. Семёнов

Приведены новые атаки из класса «угадывай и определяй» для хеш-функций вида MD4-k, $k > 39$. Описываемые атаки основаны на концепции инверсной лазейки. Для решения задач криптоанализа, ослабленных подстановками угадываемых бит, используются SAT-решатели. Задача поиска инверсной лазейки, обеспечивающей атаку с относительно малой трудоёмкостью, ставится в форме задачи минимизации специальной псевдобулевой функции. Для её решения используются три метаэвристических алгоритма: алгоритм поиска с запретами, (1+1)-FEA_β и специальный вариант генетического алгоритма. Перечисленные алгоритмы дают атаки на рассматриваемые функции с близкими оценками трудоемкости. Для функции сжатия полнораундового MD4 лучшие атаки строит генетический алгоритм.

Ключевые слова: *задача поиска прообразов криптографической хеш-функции, атаки из класса «угадывай и определяй», инверсные лазейки, SAT.*

1. О понятии инверсной лазейки

Понятие инверсной лазейки (Inverse Backdoor Set, IBS) введено в [1]. Кратко напомним его суть. Рассматривается задача обращения (поиска прообразов) произвольной функции вида

$$f : \{0,1\}^n \to \{0,1\}^m, \tag{1}$$

заданной программой (алгоритмом) M_f. Более точно, требуется по произвольному $\gamma \in$ $\in \operatorname{Range} f$ найти такой $\alpha \in \{0,1\}^n$, что $f(\alpha) = \gamma$. Подход к решению данной задачи,

[1]Работа выполнена при финансовой поддержке Российского научного фонда, проект № 16-11-10046. Грибанова И. А. поддержана стипендией Президента РФ СП-3545.2019.5.

используемый далее, относится к алгебраическому криптоанализу [2]. В соответствии с ним задачу поиска прообраза произвольного $\gamma \in \operatorname{Range} f$ можно свести к решению системы алгебраических уравнений над GF(2) либо к поиску набора, выполняющего некоторую выполнимую КНФ. Далее нам потребуется понятие шаблонной КНФ (template CNF), введённое в [3]. Шаблонная КНФ C_f строится по представлению функции f в виде схемы G_f из функциональных элементов с n входами и m выходами над произвольным полным базисом, например над $\{\wedge, \neg\}$. Для перехода от схемы G_f к C_f используются преобразования Цейтина [4].

Пусть U — множество всех булевых переменных, присутствующих в C_f; $X = \{x_1, \ldots, x_n\}$ — переменные, которые приписаны входу схемы G_f. Используем понятие подстановки произвольного значения $\lambda \in \{0,1\}$ произвольной переменной $u \in U$ в формулу C_f. Это понятие дается стандартным образом — например, в соответствии с [5]: то есть каждое вхождение переменной u в C_f заменяется на λ, после чего выполняются все возможные элементарные преобразования. В результате таких преобразований ряд не означенных ранее переменных могут принять конкретные значения. Будем говорить про такие значения, что они индуцированы соответствующей подстановкой.

Пусть $\alpha \in \{0,1\}^n$ — произвольный набор значений переменных из X. Как показано в [3], подстановка α в C_f индуцирует набор значений всех переменных из $U \setminus X$, в том числе и набор значений $\gamma = (\gamma_1, \ldots, \gamma_m)$ переменных из $Y = \{y_1, \ldots, y_m\}$, которые приписаны выходу схемы G_f. При этом имеет место $f(\alpha) = \gamma$.

Рассмотрим произвольное $B \subseteq U \setminus Y$. Зададим на $\{0,1\}^n$ равномерное распределение и свяжем с выбранным случайно набором $\alpha = (\alpha_1, \ldots, \alpha_n) \in \{0,1\}^n$ индуцированные подстановкой $x_1 = \alpha_1, \ldots, x_n = \alpha_n$ наборы значений переменных из B и из Y, которые обозначим β_α и γ_α соответственно. Обозначим через $C_f[\beta_\alpha/B, \gamma_\alpha/Y]$ КНФ, которая получена из C_f в результате подстановки в неё наборов β_α, γ_α. Зафиксируем некоторое число $t > 0$ и рассмотрим произвольный детерминированный алгоритм A решения SAT. Рассмотрим следующую величину:

$$\rho_B(t) = \frac{\#\{\alpha \in \{0,1\}^n : A(C_f[\beta_\alpha/B, \gamma_\alpha/Y]) \leqslant t\}}{2^n}. \tag{2}$$

В числителе (2) стоит число таких $\alpha \in \{0,1\}^n$, что время нахождения алгоритмом A набора, выполняющего $C_f[\beta_\alpha/B, \gamma_\alpha/Y]$, не превосходит t; в знаменателе — общее число различных α. Таким образом, $\rho_B(t)$ — вероятность следующего события: случайное α индуцирует такие β_α, γ_α, что SAT в отношении КНФ $C_f[\beta_\alpha/B, \gamma_\alpha/Y]$ решается алгоритмом A за время $\leqslant t$. Множество B называется инверсной лазейкой с параметрами $(t, \rho_B(t), s)$, где $s = |B|$. В [1] показано, как на основе инверсной лазейки с данными параметрами построить атаку из класса «угадывай и определяй» на криптографическую функцию вида (1), трудоёмкость которой равна

$$T = 2^s \cdot t \left\lceil \frac{3}{\rho_B(t)} \right\rceil.$$

Далее в [1] предлагается рассматривать задачу поиска инверсной лазейки B с относительно малой трудоёмкостью как задачу минимизации специальной функции

$$\Phi : \{0,1\}^n \to \mathbb{R}. \tag{3}$$

Напомним, что функции вида (3) называются псевдобулевыми [6]. Множество B ищется среди всевозможных подмножеств множества X. Функция (3) определяется следующим образом. Предполагается, что произвольный $\delta \in \{0,1\}^n$ задаёт конкретное B:

единицы в δ соответствуют тем и только тем переменным из X, которые входят в B. По произвольному $\delta \in \{0,1\}^n$ строится множество B, после чего генерируется случайная выборка $\alpha^1, \ldots, \alpha^N$, $\alpha^j \in \{0,1\}^n$, $j \in \{1, \ldots, N\}$. Затем наблюдаются N значений случайной величины ξ: для каждого $j \in \{1, \ldots, N\}$ данная величина принимает значение $\xi^j = 1$, если алгоритм A решает SAT для КНФ $C_f[\beta_{\alpha^j}/B, \gamma_{\alpha^j}/Y]$ за время $\leqslant t$, в противном случае $\xi^j = 0$. В роли оценки $\rho_B(t)$ используется величина $\dfrac{1}{N}\sum\limits_{j=1}^{N}\xi^j$. Соответствующее значение функции (3) определяется как

$$\Phi(\delta) = 2^{\mathrm{wt}(\delta)} \cdot t \cdot 3N / \sum_{j=1}^{N} \xi^j, \tag{4}$$

где $\mathrm{wt}(\delta)$ — вес Хэмминга вектора δ.

Заметим, что ξ — случайная величина Бернулли, $\mathrm{M}[\xi] = \rho_B(t)$, $\mathrm{D}[\xi] = \rho_B(t)(1-\rho_B(t))$. Учитывая это и используя неравенство Чебышёва [7], можно показать, что для любого $\varepsilon > 0$ имеет место

$$\mathsf{P}\left[\left|\rho_B(t) - \frac{1}{N}\sum_{j=1}^{N}\xi^j\right| \leqslant \varepsilon\right] \geqslant 1 - \frac{1}{4 \cdot \varepsilon^2 \cdot N},$$

то есть $\dfrac{1}{N}\sum\limits_{j=1}^{N}\xi^j$ позволяет оценить $\rho_B(t)$ с любой наперёд заданной точностью за счёт увеличения объёма выборки N.

2. Алгоритмы поиска инверсных лазеек

Как следует из сказанного выше, имеет смысл искать инверсные лазейки с как можно меньшим значением функции (4). Поскольку функция (4) не задана аналитически, мы можем использовать для её минимизации только эвристические алгоритмы. В настоящей работе использованы следующие алгоритмы: алгоритм из [8], относящийся к классу алгоритмов поиска с запретами [9]; т.н. «быстрый эволюционный алгоритм» (1+1)–FEA_β [10], а также специальный вариант генетического алгоритма [11]. Дадим краткое описание этих алгоритмов.

Алгоритм поиска с запретами (далее — TS от Tabu Search) из [8] — это вариант локального поиска, который хранит все пройденные точки $\{0,1\}^n$ в специальных списках и запрещает повторно вычислять значения функции (4) в точке, в которой эта функция уже вычислялась. Многократное вычисление значений (4) в одних и тех же точках приводит к замедлению поиска, поскольку каждое такое вычисление требует существенного времени. К тому же, как показано в [9], такая стратегия позволяет алгоритму выходить из точек локального минимума (полнота при отсутствии ограничений по памяти).

Алгоритм (1+1)–FEA_β, описанный в [10], представляет собой усложнённый вариант известного эволюционного алгоритма (1+1)–EA [12]. Идея, лежащая в основе (1+1)–FEA_β, состоит в том, чтобы использовать переменную вероятность мутации. В классическом (1+1)–EA вероятность мутации, то есть изменения произвольного бита в рассматриваемом слове $\alpha \in \{0,1\}^n$ на противоположный, постоянна и равна $1/n$. Если α — исходное слово из $\{0,1\}^n$, а α' — результат случайной мутации α в соответствии с (1+1)–EA, то математическое ожидание случайной величины $\mathrm{H}(\alpha, \alpha')$ (расстояния Хэмминга между α и α') есть $\mathrm{M}[\mathrm{H}(\alpha, \alpha')] = 1$. Это свойство очень важно [13], поскольку оно означает, что в среднем данный алгоритм ведёт себя похожим на стандартный локальный поиск образом и соответственно имеет возможность приспосабли-

ваться под «ландшафт» рассматриваемой функции. С другой стороны, этот алгоритм с ненулевой вероятностью переходит в произвольную точку гиперкуба $\{0,1\}^n$. Однако (1+1)–EA имеет крайне плохую верхнюю оценку сложности в смысле меры, введённой в [14], — конкретно, данная оценка имеет вид n^n и, таким образом, (1+1)–EA существенно менее эффективен (в данном смысле), чем простой случайный поиск. В алгоритме (1+1)–FEA_β вероятность мутации зависит от поведения специальным образом определённой случайной величины. В зависимости от значений параметра β алгоритм (1+1)–FEA_β может демонстрировать различные сочетания основных свойств. Наиболее интересным с практических позиций является значение $\beta = 3$, так как в этом случае верхняя оценка сложности (1+1)–FEA_3 в смысле [14] есть $\Theta(n^3 \cdot 2^n)$, притом что $\mathrm{M}[\mathrm{H}(\alpha, \alpha')] \approx \dfrac{\zeta(2)}{\zeta(3)} \approx 1{,}3685$ (здесь ζ — дзета-функция Римана).

Ещё один алгоритм, использованный для минимизации (4), — это специальный случай генетического алгоритма, который описан в [11] (далее — GA от Genetic Algorithm). В данном алгоритме по набору векторов $P = \{\lambda^1, \ldots, \lambda^M\}$, $\lambda^i \in \{0,1\}^n$, $i \in \{1, \ldots, M\}$, строится новый набор $\tilde{P} = \{\tilde{\lambda}^1, \ldots, \tilde{\lambda}^M\}$ в соответствии с несколькими базовыми концепциями теории генетических алгоритмов. Начальный набор из M векторов строится либо случайным образом, либо как результат работы других алгоритмов, например (1+1)–EA. Часть наборов в $\tilde{P}$ состоит из лучших по значению целевой функции элементов P. Другая часть наборов в $\tilde{P}$ есть результат стандартных (1+1)–EA мутаций над несколькими наборами, случайно выбранными из P. Наконец, оставшиеся наборы из $\tilde{P}$ получаются в результате операции двухточечного кроссовера [15] над наборами, случайно выбираемыми из P. Для каждого элемента $\tilde{P}$ вычисляется значение функции (4).

3. Атаки на основе инверсных лазеек на функции вида MD4-k, $k > 39$

Везде далее под MD4-k понимается функция сжатия, задаваемая первыми k шагами известного алгоритма хеширования MD4 [16]. В основе предлагаемых атак лежит идея дополнения уравнений криптоанализа функций вида MD4-k ослабляющими ограничениями. Данная идея высказана Г. Доббертином в [17] и адаптирована к использованию SAT-решателей в [18]. В [19] описан алгоритм, позволяющий генерировать ослабляющие ограничения «типа Доббертина» автоматически. С использованием данного алгоритма построена рекордная по трудоёмкости атака на функцию MD4-39. В дальнейшем при помощи подхода из [19] были построены новые атаки для функций вида MD4-k до $k = 48$ включительно. В частности, атака такого типа на полнораундовую функцию сжатия MD4, представленная в [20], показывает, что данная функция не обладает свойствами случайного оракула. Атаки, описанные в [19 – 21], эксплуатируют общую идею перехода от задачи обращения функции MD4-k к задаче обращения вспомогательных функций вида

$$g^{\lambda}_{\text{MD4-}k} : \{0,1\}^d \to \{0,1\}^{128}, \tag{5}$$

таких, что $d \ll 512$. Через λ в (5) обозначен булев вектор, задающий ослабляющие ограничения «типа Доббертина». Генерируя при помощи алгоритма из [19] различные λ, можно строить нетривиальные атаки на функции вида MD4-k.

В рамках настоящей работы мы использовали описанные в п. 2 метаэвристические алгоритмы для поиска инверсных лазеек в задачах обращения следующих функций:

$$\begin{gathered} g^{\lambda_1}_{\text{MD4-40}} : \{0,1\}^{128} \to \{0,1\}^{128}, \quad g^{\lambda_1}_{\text{MD4-48}} : \{0,1\}^{128} \to \{0,1\}^{128}, \\ g^{\lambda_3}_{\text{MD4-48}} : \{0,1\}^{96} \to \{0,1\}^{128}, \quad g^{\lambda_5}_{\text{MD4-48}} : \{0,1\}^{64} \to \{0,1\}^{128}. \end{gathered} \tag{6}$$

Более подробную информацию о векторах λ_1, λ_3 и λ_5 и перечисленных функциях можно найти в [20]. Заметим, что если $\chi \in \{0,1\}^{128}$ — значение любой из функций (6) и x — прообраз χ в смысле этой функции, то от x можно эффективно перейти к MD4-прообразу χ.

Результаты построения инверсных лазеек описанными алгоритмами и оценки трудоёмкости соответствующих атак из класса «угадывай и определяй» в применении к функциям (6) приведены в таблице.

Алгоритм	$g^{\lambda_1}_{\text{MD4-40}}$	$g^{\lambda_1}_{\text{MD4-48}}$	$g^{\lambda_3}_{\text{MD4-48}}$	$g^{\lambda_5}_{\text{MD4-48}}$
TS 1 ($t = 100$)	$\langle 20\rangle$ 4,4 e+10	$\langle 100\rangle$ 4,5 e+34	$\langle 66\rangle$ 1,2 e+24	$\langle 28\rangle$ 7,8 e+12
TS 2 ($t = 200$)	$\langle 15\rangle$ 2,4 e+9	$\langle 98\rangle$ 2,7 e+34	$\langle 63\rangle$ 7,2 e+23	–
(1+1)–FEA_3 1 ($t = 100$)	$\langle 23\rangle$ 2,8 e+11	$\langle 104\rangle$ 2,1 e+35	$\langle 65\rangle$ 6,7 e+23	$\langle 28\rangle$ 1,1 e+13
(1+1)–FEA_3 2 ($t = 200$)	$\langle 20\rangle$ 9,1 e+10	$\langle 100\rangle$ 1,1 e+35	$\langle 64\rangle$ 5,7 e+23	–
GA 1 ($t = 100$)	$\langle 22\rangle$ 1,7 e+11	$\langle 100\rangle$ 2,0 e+34	$\langle 63\rangle$ 3,6 e+23	$\langle 27\rangle$ 3,5 e+12
GA 2 ($t = 200$)	$\langle 21\rangle$ 1,5 e+11	–	–	–

К о м м е н т а р и и к т а б л и ц е. В первом столбце приведено название алгоритма. В ряде случаев процесс поиска лазейки разбивался на два этапа: на первом этапе использовалось значение $t = 100$ с (см. (4)), затем с лучшей найденной точки запускался этот же алгоритм с параметром $t = 200$ с. В последующих столбцах приведено число переменных в соответствующих лазейках и значения функции (4) для этих лазеек. Каждое такое значение даёт оценку времени выполнения атаки из класса «угадывай и определяй» в секундах для соответствующей функции вида (6) на одном ядре используемого процессора (в нашем случае — на одном ядре AMD Opteron 6276). По результатам экспериментов можно сделать вывод о том, что все сравниваемые алгоритмы дают атаки с близкими оценками трудоёмкости. Для полнораундовой версии функции сжатия MD4 атаки с наименьшей трудоёмкостью строит генетический алгоритм с $M = 10$.

Все вычислительные эксперименты по поиску инверсных лазеек для функций вида (6) проводились на вычислительном кластере «Академик В. М. Матросов» Иркутского суперкомпьютерного центра [22].

ЛИТЕРАТУРА

1. *Semenov A., Zaikin O., Otpuschennikov I., et al.* On cryptographic attacks using backdoors for SAT // Proc. 32nd AAAI Conf. 2018. P. 6641–6648.
2. *Bard G.* Algebraic Cryptanalysis. Springer Publishing Company, Inc., 2009. 356 p.
3. *Semenov A., Otpuschennikov I., Gribanova I., et al.* Translation of algorithmic descriptions of discrete functions to SAT with application to cryptanalysis problems // Log. Methods Comput. Sci. 2020. V. 16. Iss. 1. P. 29:1–29:42.
4. *Цейтин Г. С.* О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ АН СССР. 1968. Т. 8. С. 234–259.
5. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983. 360 с.
6. *Boros E. and Hammer P. L.* Pseudo-Boolean optimization // Discrete Appl. Math. 2002. V. 123 (1-3). P. 155–225.
7. *Феллер У.* Введение в теорию вероятностей и ее приложения. Т. 1. М.: Мир, 1964. 500 с.
8. *Semenov A. and Zaikin O.* Algorithm for finding partitionings of hard variants of Boolean satisfiability problem with application to inversion of some cryptographic functions // SpringerPlus. 2016. V. 5 (1). P. 1–16.

9. *Glover F. and Laguna M.* Tabu Search. Norwell: Kluwer Academic Publishers, 1997. 401 p.
10. *Doerr B., Le H., Makhmara R., et al.* Fast genetic algorithms // Proc. GECCO'17. 2017. P. 777–784.
11. *Pavlenko A., Semenov A., and Ulyantsev V.* Evolutionary computation techniques for constructing SAT-based attacks in algebraic cryptanalysis // LNCS. 2019. V. 11454. P. 237–253.
12. *Muhlenbein H.* How genetic algorithms really work: Mutation and hill climbing // Proc. PPSN-II. 1992. P. 15–26.
13. *Wegener I.* Theoretical aspects of evolutionary algorithms // ICALP 2001. LNCS. 2001. V. 2076. P. 64–78.
14. *Droste S., Jansen T., and Wegener I.* On the analysis of the (1+1) evolutionary algorithm // Theor. Comput. Sci. 2002. V. 276 (1–2). P. 51–81.
15. *Luke S.* Essentials of Metaheuristics. Second Edition. 2015. 261 p. `https://cs.gmu.edu/~sean/book/metaheuristics/Essentials.pdf`.
16. *Rivest R. L.* The MD4 message digest algorithm // CRYPTO'90. LNCS. 1990. V. 537. P. 303–311.
17. *Dobbertin H.* The first two rounds of MD4 are not one-way // FSE 1998. LNCS. 1998. V. 1372. P. 284–292.
18. *De D., Kumarasubramanian A., and Venkatesan R.* Inversion attacks on secure hash functions using SAT Solvers // FSE 2007. LNCS. 2007. V. 4501. P. 377–382.
19. *Gribanova I. and Semenov A.* Using automatic generation of relaxation constraints to improve the preimage attack on 39-step MD4 // Proc. 41st Intern. Convention MIPRO 2018. Opatija, 2018. P. 1174–1179.
20. *Грибанова И. А., Семёнов А. А.* Об аргументации отсутствия свойств случайного оракула у некоторых криптографических хеш-функций // Прикладная дискретная математика. Приложение. 2019. № 12. С. 95–98.
21. *Gribanova I. A. and Semenov A. A.* Parallel guess-and-determine preimage attack with realistic complexity estimation for MD4-40 cryptographic hash function // Труды XIII Междунар. конф. «Параллельные вычислительные технологии», Калининград, 02–04 апреля 2019. С. 8–18.
22. Иркутский суперкомпьютерный центр СО РАН. `http://hpc.icc.ru`.

УДК 519.7 DOI 10.17223/2226308X/13/38

ПРИМЕНЕНИЕ SAT-РЕШАТЕЛЕЙ ДЛЯ ПОСТРОЕНИЯ БУЛЕВЫХ ФУНКЦИЙ С ЗАДАННЫМИ КРИПТОГРАФИЧЕСКИМИ СВОЙСТВАМИ[1]

А. Е. Доронин, К. В. Калгин

Представлен подход к решению некоторых криптографических задач, основанный на их сведении к классической задаче о выполнимости и последующем использовании SAT-решателей. Построены формулы, определяющие условия взаимной однозначности и дифференциальной равномерности векторной булевой функции.

Ключевые слова: *SAT-решатели, криптография, булевы функции.*

[1]Работа выполнена при поддержке РФФИ (проект № 18-07-01394) и Лаборатории криптографии JetBrains Research.

В настоящее время SAT-решатели используются для решения криптографических задач разного типа. Например, проведён криптоанализ асимметричной криптосистемы RSA [1], в результате которого удалось факторизовать числа до 417 бит; выполнен криптоанализ шифра Trivium и его модификаций [2]. В [3] представлена гомоморфная криптосистема с открытым ключом, основанная на SAT-задаче. С помощью SAT-решателей успешно проверяется обратимость векторных булевых функций [4].

В данной работе предлагается использование SAT-решателей в задачах построения криптографических булевых функций и проверки эквивалентности двух булевых функций. Для получения набора булевых формул использованы следующие понятия и свойства.

Определение 1. Векторная булева функция $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ называется *взаимно однозначной*, если она инъективна и сюръективна, то есть одновременно выполняются следующие условия:

1) $\forall x' \in \mathbb{Z}_2^n \; \forall x'' \in \mathbb{Z}_2^n \; (x' \neq x'' \to F(x') \neq F(x''))$;
2) $\forall y \in \mathbb{Z}_2^n \; \exists x \in \mathbb{Z}_2^n \; (F(x) = y)$.

Определение 2. Векторная булева функция $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ является *дифференциально δ-равномерной*, если для любого ненулевого $a \in \mathbb{Z}_2^n$ и произвольного $b \in \mathbb{Z}_2^n$ уравнение $F(x) \oplus F(x \oplus a) = b$ имеет не более δ решений.

Условия, фигурирующие в определениях, представляются в виде КНФ и подаются на вход SAT-решателя. В результате его работы происходит означивание переменных таким образом, чтобы формулы были истинными, а следовательно, условия выполнялись.

Векторные булевы функции были закодированы в двух представлениях:

1) В *разреженном* представлении используется 2^{2n} переменных $f_{x,y}$, из которых 2^n равны 1, остальные равны 0: $f_{x,y} = 1 \iff F(x) = y$, где $x, y \in \mathbb{Z}_2^n$.
2) В *плотном* представлении используется $n2^n$ переменных $fb_{x,k}$: $fb_{x,k} = 1 \iff F_k(x) = 1$, где $F(x) = (F_0(x), F_2(x), \ldots, F_{n-1}(x))$, $F_k : \mathbb{Z}_2^n \to \mathbb{Z}_2$, $k = 0, \ldots, n-1$; $x \in \mathbb{Z}_2^n$.

Для записи условий на переменные $f_{x,y}$ и $fb_{x,k}$ понадобятся следующие вспомогательные переменные:

— $fbq_{x,y,k} = 1 \iff fb_{x,k} \neq fb_{y,k}$, где $k = 0, \ldots, n-1$; $x, y \in \mathbb{Z}_2^n$;
— $d_{x,a,b} = 1 \iff F(x) \oplus F(x \oplus a) = b$, где $x, a, b \in \mathbb{Z}_2^n$;
— $de_{x,y,a} = 1 \iff F(x) \oplus F(x \oplus a) = F(y) \oplus F(y \oplus a)$, где $x, y, a \in \mathbb{Z}_2^n$;
— $dbq_{x,y,a,k} = 1 \iff fbq_{x,x\oplus a,k} = fbq_{y,y\oplus a,k}$, где $k = 0, \ldots, n-1$, $x, y, a \in \mathbb{Z}_2^n$.

В КНФ эти зависимости записываются следующим образом:

$$\mathbf{SoP^D}(fb, fbq) = \bigwedge_{x,y,k} (fbq_{x,y,k} \vee fb_{x,k} \vee \overline{fb_{y,k}}) \wedge (fbq_{x,y,k} \vee \overline{fb_{x,k}} \vee fb_{y,k}) \wedge$$
$$\wedge (\overline{fbq_{x,y,k}} \vee fb_{x,k} \vee fb_{y,k}) \wedge (\overline{fbq_{x,y,k}} \vee \overline{fb_{x,k}} \vee \overline{fb_{y,k}});$$
$$\mathbf{SpDen}(f, fb) = \bigwedge_{x,y,k} (\overline{f_{x,y}} \vee fb_{x,k}) \wedge (f_{x,y} \vee \overline{fb_{x,0}^{y_0}} \vee \ldots \vee \overline{fb_{x,n-1}^{y_{n-1}}});$$
$$\mathbf{Der^S}(f, d) = \bigwedge_{b,a,z,x} (f_{x,z} \vee f_{x\oplus a,z\oplus b} \vee \overline{d_{x,a,b}}) \wedge (f_{x,z} \vee \overline{f_{x\oplus a,z\oplus b}} \vee \overline{d_{x,a,b}}) \wedge$$
$$\wedge (\overline{f_{x,z}} \vee f_{x\oplus a,z\oplus b} \vee \overline{d_{x,a,b}}) \wedge (\overline{f_{x,z}} \vee \overline{f_{x\oplus a,z\oplus b}} \vee d_{x,a,b});$$

$$\mathbf{SoPEq^D}(fbq, dbq) = \bigwedge_{a,x,y,k} (dbq_{x,y,a,k} \vee fbq_{x,x\oplus a,k} \vee fbq_{y,y\oplus a,k}) \wedge$$
$$\wedge (dbq_{x,y,a,k} \vee \overline{fbq_{x,x\oplus a,k}} \vee \overline{fbq_{y,y\oplus a,k}}) \wedge (\overline{dbq_{x,y,a,k}} \vee fbq_{x,x\oplus a,k} \vee \overline{fbq_{y,y\oplus a,k}}) \wedge$$
$$\wedge (\overline{dbq_{x,y,a,k}} \vee \overline{fbq_{x,x\oplus a,k}} \vee fbq_{y,y\oplus a,k});$$
$$k = 0, \ldots, n-1;\ x, y, z, a, b \in \mathbb{Z}_2^n.$$

Свойства из определений 1 и 2 можно записать следующими формулами.

Теорема 1. Переменные $f_{x,y}$ задают функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда следующая формула является истинной:

$$\mathbf{F^S}(f) = \bigwedge_{x\in\mathbb{Z}_2^n} \Big(\bigwedge_{\substack{y',y''\in\mathbb{Z}_2^n \\ y'<y''}} \overline{f_{x,y'}} \vee \overline{f_{x,y''}} \Big) \wedge \bigwedge_{x\in\mathbb{Z}_2^n} \Big(\bigvee_{y\in\mathbb{Z}_2^n} f_{x,y} \Big). \tag{1}$$

Формула (1) состоит из $2^{3n-1} - 2^{2n-1}$ дизъюнкций длины 2 и 2^n дизъюнкций длины 2^n.

Теорема 2. Переменные $f_{x,y}$ задают взаимно однозначную векторную булеву функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда следующая формула является истинной:

$$\mathbf{P^S}(f) = \bigwedge_{y\in\mathbb{Z}_2^n} \Big(\bigwedge_{\substack{x',x''\in\mathbb{Z}_2^n \\ x'<x''}} \overline{f_{x',y}} \vee \overline{f_{x'',y}} \Big) \wedge \mathbf{F^S}(f). \tag{2}$$

Формула (2) состоит из $2^{3n} - 2^{2n}$ дизъюнкций длины 2 и 2^n дизъюнкций длины 2^n.

Теорема 3. Переменные $fb_{x,k}$ и $fbq_{x,y,k}$ задают взаимно однозначную векторную булеву функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда следующая формула является истинной:

$$\mathbf{P^D_{sum}}(fb, fbq) = \bigwedge_{x,y\in\mathbb{Z}_2^n} \bigvee_k fbq_{x,y,k} \wedge \mathbf{SoP^D}(fb, fbq). \tag{3}$$

В формуле (3) содержится $n(2^{2n} - 2^n)$ дизъюнкций длины 3 и $2^{2n} - 2^n$ дизъюнкций длины n.

Теорема 4. Переменные $f_{x,y}$ и $fb_{x,k}$ задают взаимно однозначную векторную булеву функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда следующая формула является истинной:

$$\mathbf{P^D_{sparse}}(f, fb) = \bigwedge_x \bigvee_{y\neq x} f_{x,y} \wedge \mathbf{SpDen}(f, fb). \tag{4}$$

В формуле (4) содержится по $n(2^{2n} - 2^n)$ дизъюнкций длины 2 и n и 2^n дизъюнкций длины 2^n.

Теорема 5. Переменные $f_{x,y}$ и $d_{x,a,b}$ задают APN-функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда выполняются условия теоремы 1 и следующая формула является истинной:

$$\mathbf{APN^S}(f, d) = \mathbf{Der^S}(f, d) \wedge \bigwedge_{\substack{b\neq 0, a\neq 0, \\ x,y\neq x}} (\overline{d_{x,a,b}} \vee \overline{d_{y,a,b}}). \tag{5}$$

В формуле (5) содержится порядка 2^{4n} дизъюнкций длины 3 и 2.

Теорема 6. Переменные $f_{x,y}$ и $d_{x,a,b}$ задают APN-функцию $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ тогда и только тогда, когда следующая формула является истинной:

$$\mathbf{APN^D}(fb, fbq, dbq) = \mathbf{SoPEq^D}(fbq, dbq) \wedge \mathbf{SoP^D}(fb, fbq) \wedge \bigwedge_{a,x,y} \bigvee_k \overline{dbq_{x,y,a,k}}. \tag{6}$$

В формуле (6) содержится порядка 2^{3n} дизъюнкций длины 3 и n.

На основе полученных формул генерируется входной файл для SAT-решателя. Формулы можно также использовать для тестирования работы новых SAT-решателей, созданных для криптографических задач.

ЛИТЕРАТУРА

1. *Огородников Ю. Ю.* Комбинированная атака на алгоритм RSA с использованием SAT-подхода // Динамика систем, механизмов и машин. Омск: ОмГТУ, 2016. С. 276–284.
2. *Заикин О. С., Отпущенников И. В., Семёнов А. А.* Оценки стойкости шифров семейства Trivium к криптоанализу на основе алгоритмов решения проблемы булевой выполнимости // Прикладная дискретная математика. Приложение. 2016. № 9. С. 46–48.
3. *Schmittner S. E.* A SAT-based Public Key Cryptography Scheme. IACR Cryptol. ePrint Arch. 2015. `https://eprint.iacr.org/2015/771.pdf`.
4. *Wille R., Lye A., and Niemann P.* Checking reversibility of Boolean functions // LNCS. 2016. V. 9720. P. 322–337.

УДК 519.688 DOI 10.17223/2226308X/13/39

О ВЫЧИСЛЕНИИ СИСТЕМЫ ПЕРЕПИСЫВАЮЩИХ ПРАВИЛ В КОНЕЧНОЙ ГРУППЕ

А. А. Кузнецов

Представлен алгоритм, определяющий переписывающую систему конечной группы, заданной фиксированным порождающим множеством. Необходимым условием эффективной реализации алгоритма является наличие быстрой процедуры умножения элементов в группе. Такой групповой операцией может быть композиция подстановок, умножение матриц, вычисление полиномов Холла и т. д. Алгоритм был применён для исследования переписывающих систем в конечных двупорождённых группах периода 5.

Ключевые слова: *система переписывающих правил, группа Бернсайда.*

Решение некоторых задач теории кодирования и криптографии сводится к исследованию подходящих графов Кэли, например открытая проблема эффективного восстановления вершин в графе Хэмминга [1].

Поиск кратчайших путей в графах Кэли является труднорешаемой проблемой, поэтому исследователям приходится идти на различные уловки и приёмы, чтобы получить решение за приемлемое время. Например, в [2] сначала определяют автоматическую структуру группы, которая порождает соответствующий граф Кэли. Автоматическая структура группы состоит из конечных автоматов специального вида [3]. Для их вычисления требуется определить множество соотношений в группе, используя известный алгоритм Кнута — Бендикса [4].

Зачастую алгоритм Кнута — Бендикса работает недопустимо долго, например в конечных группах, заданных коммутаторными соотношениями. В этом случае разворачивание коммутаторных соотношений приводит к очень длинным словам, что катастрофически замедляет работу алгоритма.

Настоящая работа представляет собой попытку устранить указанный недостаток. Остановимся подробнее на основных определениях.

Пусть $G = \langle X \rangle$ — конечная группа, порождённая упорядоченным множеством $X = \{x_1 \prec x_2 \prec \ldots \prec x_m\}$, которое также называют алфавитом. Множество всех

слов (строк) над алфавитом X будем обозначать X^*. Пусть $w = x_1x_2\dots x_l$ — слово над X и $|w| = l$ — его длина. На множестве X^* также определим отношение порядка. Пусть v и w — два произвольных слова в алфавите X. Тогда $v \prec w$, если $|v| < |w|$, а в случае равенства длин меньшее слово определяется согласно введённому лексикографическому порядку на порождающих. Если необходимо подчеркнуть, что строка $v \in X^*$ соответствует элементу $g \in G$, то будем писать v_g. Строку v будем называть минимальным словом элемента g, если для всех других $w \in X^*$, таких, что $v_g = w_g$, выполняется $v \prec w$. Очевидно, что каждому $g \in G$ соответствует уникальное минимальное слово. Единице группы e соответствует пустое слово ε: $|\varepsilon| = 0$.

Пусть R — система переписывающих правил (переписывающая система), состоящая из множества пар вида (u, v), где $u_g = v_g$ и $u \succ v$ [4]. При этом слово u называют левой стороной правила, а строку v — правой. Иногда правила записывают в виде $u \to v$. Действие системы R над некоторым словом w означает осуществление замен вида $xuy \to xvy$ до тех пор, пока не будет получено несократимое относительно R слово w', т. е. $R(w) = w'$.

Если изменение порядка применения правил не влияет на конечный результат, то R называют *конфлюэнтной*.

Переписывающую систему R называют *несократимой*, если для любой пары $(u, v) \in R$ выполняется $R'(u) = u$ и $R'(v) = v$, где $R' = R \setminus \{(u, v)\}$.

Алгоритм 1 определяет переписывающую систему конечной группы $G = \langle X, \circ\rangle$. Необходимым условием эффективной реализации алгоритма является наличие быстрой процедуры умножения элементов в группе. Например, групповой операцией $\circ$ может быть композиция подстановок, умножение матриц, вычисление полиномов Холла и т. д.

Алгоритм 1. $R =$ `RewritingSystem`$(G, X, \circ)$

Вход: $G = \langle X, \circ\rangle$.
Выход: система переписывающих правил R группы G.

$P_0 := \{\varepsilon\}$ — множество минимальных слов.
$K_0 := \{(e, \varepsilon)\}$ — словарь вида (элемент группы, его минимальное слово).
$R := \varnothing$.
Для всех $i = 1, 2, \dots, \infty$:
  $K_i := K_{i-1}$, $P_i := \varnothing$.
  Для всех $x \in X$ и $p \in P_{i-1}$:
    $u := xp$ — конкатенация слов,
    $g := x \circ p$ — групповое умножение.
    Если $g \in K_i$, **то**
      если $R(u) = u$, **то** $v := K_i[g]$, добавить (u, v) в R,
    иначе
      добавить u в P_i, добавить (g, u) в K_s.
  Если $P_i = \varnothing$, **то**
    Вернуть R.

Теорема 1. Пусть R — система переписывающих правил, полученная при помощи алгоритма 1, тогда R конфлюэнтна и несократима.

Рассмотрим примеры. Пусть $B_0(2, 5) = \langle a_1, a_2\rangle$ — наибольшая конечная двупорождённая бернсайдова группа периода 5, порядок которой равен 5^{34} [5]. Для каж-

дого элемента данной группы существует уникальное коммутаторное представление вида $a_1^{\alpha_1} \cdot a_2^{\alpha_2} \cdot \ldots \cdot a_{34}^{\alpha_{34}}$, где $\alpha_i \in \mathbb{Z}_5$, $i = 1, 2, \ldots, 34$. Здесь a_1 и a_2 — порождающие элементы $B_0(2,5)$, $a_3, \ldots, a_{34}$ — коммутаторы, которые вычисляются рекурсивно через a_1 и a_2. Определим фактор-группу группы $B_0(2,5)$ следующего вида: $B_k = B_0(2,5)/\langle a_{k+1}, \ldots, a_{34}\rangle$. Очевидно, что $|B_k| = 5^k$.

Пусть R_k — переписывающая система группы B_k. На рис. 1 представлены графики роста R_k для минимального порождающего множества $X = \langle a_1, a_2\rangle$, а также симметричного $Y = \langle a_1, a_2, a_1^{-1}, a_2^{-1}\rangle$.

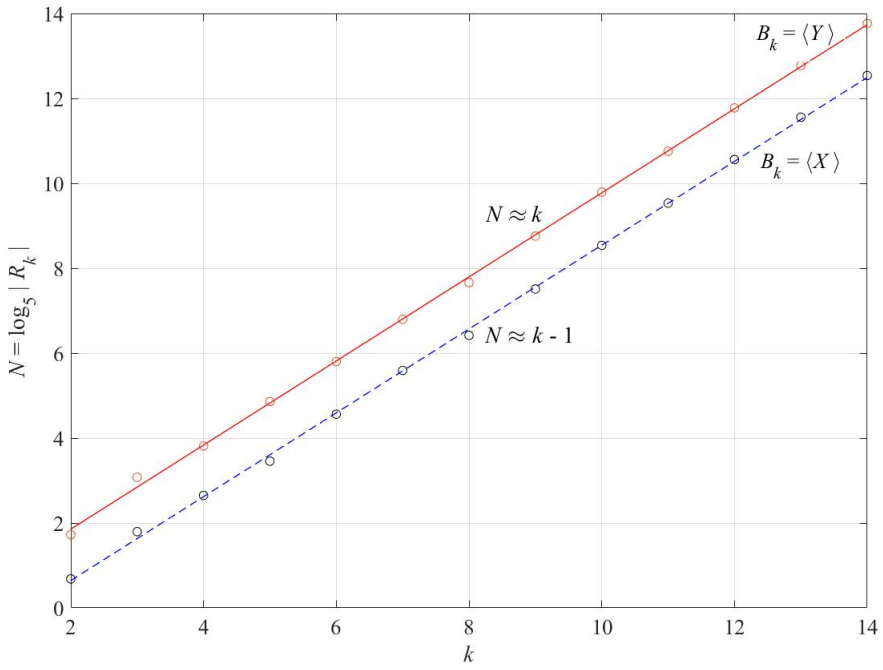

Рис. 1. Графики роста соотношений в B_k

ЛИТЕРАТУРА

1. *Константинова Е. В.* Комбинаторные задачи на графах Кэли. Новосибирск: НГУ, 2010. 110 с.
2. *Camelo M., Papadimitriou D., Fàbrega L., and Vilà P.* Efficient routing in Data Center with underlying Cayley graph // Proc. 5th Workshop Complex Networks CompleNet. 2014. P. 189–197.
3. *Epstein D., Paterson M., Cannon J., et al.* Word Processing in Groups. Boston: Jones and Barlett Publ., 1992. 330 p.
4. *Sims C.* Computation with Finitely Presented Groups. Cambridge: Cambridge University Press, 1994. 628 p.
5. *Havas G., Wall G., and Wamsley J.* The two generator restricted Burnside group of exponent five // Bull. Austral. Math. Soc. 1974. No. 10. P. 459–470.

УДК 519.7 DOI 10.17223/2226308X/13/40

КОМПАКТНЫЙ ТРАНСЛЯТОР АЛГОРИТМОВ В БУЛЕВЫ ФОРМУЛЫ ДЛЯ ПРИМЕНЕНИЯ В КРИПТОАНАЛИЗЕ[1]

Д. А. Софронова, К. В. Калгин

Представлен транслятор, позволяющий преобразовывать описание криптографической задачи (криптоанализ шифра или хэш-функции, поиск APN-функций) в КНФ. В дальнейшем SAT-решатель устанавливает истинность формулы и находит набор, выполняющий КНФ. Отличительные особенности данной разработки — универсальность, малый объём исходного кода (300 строк C++), легко модифицируемая и расширяемая реализация.

Ключевые слова: *криптоанализ, SAT-решатель, атака «угадай-и-вычисли».*

В основе одного из методов анализа симметричных шифров лежит использование SAT-решателей. По алгоритму, задающему криптографическую функцию, строится КНФ. Если для данной КНФ удаётся найти выполняющий набор, то имеем решение задачи криптоанализа. SAT-задача — задача определения выполнимости логической формулы [1]. SAT-решатель — программа, которая ищет набор значений переменных, на котором формула истинна. Известно, что эта задача NP-полная. Несмотря на это, для множества практических задач SAT-решатели определяют выполнимость формул с тысячами переменных за приемлемое время. Для проведения криптоанализа с помощью SAT-решателя необходим механизм представления криптографических алгоритмов в виде КНФ в формате DIMACS.

На данный момент существует несколько разработок, позволяющих на выходе получать КНФ. Приведём краткое описание двух разработок, специализирующихся на криптоанализе шифров — Grain of Salt [2] и Transalg [3].

Transalg универсален и позволяет сводить к задаче выполнимости не только криптографические задачи, но и некоторые задачи биоинформатики. Шифр описывается на специальном си-подобном языке, после чего строится КНФ. В настоящее время при помощи Transalg получены SAT-кодировки многих симметричных алгоритмов, а также хэш-функций [4]. Являясь полноценным транслятором, Transalg анализирует текст описания с помощью лексического, синтаксического и семантического анализаторов, что делает его достаточно сложным для модификации и расширения.

Grain of Salt (GoS) — программный комплекс описания поточных шифров на базе регистров сдвига и последующего автоматического проведения атаки «угадай-и-вычисли», который разработал автор cryptominisat [5] M. Soos. Данный вариант хорошо оптимизирован с помощью карт Карно, поэтому выходная КНФ имеет меньший размер по сравнению с КНФ, полученной без оптимизации. Построены SAT-кодировки шифров Grain, Trivium, Bivium, Crypto1 и Hitag2 [2].

В данной работе представлен программный комплекс, универсальный, легко расширяемый, простой и понятный для пользователей (в том числе на уровне реализации). Под криптографическими задачами далее подразумеваем не только задачи анализа шифров и хэш-функций, но и задачи поиска APN-функций, определения EA-эквивалентности булевых и векторных функций.

[1]Работа выполнена при поддержке Математического центра в Академгородке (г. Новосибирск), соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2019-1613, и Лаборатории криптографии JetBrains Research.

Основная идея заключается в том, что криптографическая задача (алгоритм или множество ограничений) описывается на языке C++ с использованием специальных классов varBool и varInt, у которых переопределены все операторы. Полиморфизм в C++ позволяет переопределить работу операторов для новых типов так, что при выполнении некоторых действий над данными происходит формирование КНФ (в зависимости от операций добавляются разные конструкции) или реальное исполнение алгоритма. Через указание параметров при компиляции можно получить реализацию исходного алгоритма или генератор, получающий на выходе КНФ. Кроме того, есть возможность задать значения определённых переменных varBool до генерации КНФ, например для частичного задания битов ключа при проведении атаки «угадай-и-вычисли». При использовании C-интерфейса SAT-решателя cryptominisat можно запускать решатель без промежуточной записи КНФ в файл. Работа программы построена на операциях, обрабатывающих новые типы и неявно формирующих КНФ на основе логики операций. Немаловажным плюсом является то, что большинство шифров описаны на языке C. Построение задачи криптоанализа таких шифров легко осуществляется в проекте заменой типов данных в коде. Аналогичным образом преобразуются алгоритмы, описанные на языке TransAlg. Программа является гибкой, использование возможностей языка C++ (циклы, условные операторы, классы, шаблоны) позволяет описывать алгоритмы разной сложности. На данном этапе с использованием транслятора и описанных в нём механизмов регистров сдвига построены SAT-кодировки шифров, описанных в [2], а также генератор A5/1.

ЛИТЕРАТУРА

1. *Otpuschennikov I., Semenov A., Gribanova I., et al.* Encoding cryptographic functions to SAT using TRANSALG system // Proc. ECAI'16. IOS Press, 2016. P. 1594–1595.
2. *Biere A., Heule M., Maaren H., and Walsh T.* Handbook of Satisfiability. IOS Press, 2009. 966 p.
3. *Semenov A., Otpuschennikov I., Gribanova I., et al.* Translation of algorithmic descriptions of discrete functions to SAT with application to cryptanalysis problems // Log. Methods Comput. Sci. 2020. V. 16. Iss. 1. P. 29:1–29:42.
4. *Soos M., Nohl K., and Castelluccia C.* Extending SAT solvers to cryptographic problems // LNCS. 2009. V. 5584. P. 244–257.
5. *Soos M.* Grain of salt — an automated way to test stream ciphers through SAT solvers // Tools. 2010. V. 10. P. 131–144.

СВЕДЕНИЯ ОБ АВТОРАХ

АБРОСИМОВ Михаил Борисович — доктор физико-математических наук, заведующий кафедрой Саратовского национального исследовательского государственного университета им. Н. Г. Чернышевского, г. Саратов. E-mail: **mic@rambler.ru**

АГИЕВИЧ Сергей Валерьевич — кандидат физико-математических наук, заведующий НИЛ проблем безопасности информационных технологий НИИ прикладных проблем математики и информатики, Белорусский государственный университет, г. Минск. E-mail: **agievich@bsu.by**

АНТОНОВ Кирилл Валентинович — студент ИМИТ ИГУ, г. Иркутск. E-mail: **aknitr@mail.ru**

БЕЛОУСОВА Алина Александровна — младший научный сотрудник Института математики им. С. Л. Соболева СО РАН, студентка Новосибирского государственного университета, г. Новосибирск. E-mail: **alinkabel18@gmail.com**

БЕСПАЛОВ Михаил Сергеевич — доктор физико-математических наук, доцент, профессор Владимирского государственного университета, г. Владимир. E-mail: **bespalov@vlsu.ru**

БОБРОВСКИЙ Дмитрий Александрович — студент магистратуры Финансового университета при Правительстве РФ, системный аналитик ООО «Код безопасности», г. Москва.
E-mail: **dabobrovskiy@gmail.com**

БОНИЧ Татьяна Андреевна — магистрантка Новосибирского государственного университета, исследователь Лаборатории криптографии JetBrains Research, г. Новосибирск.
E-mail: **t.bonich@g.nsu.ru**

ВЕДУНОВА Марина Викторовна — студентка Уральского государственного университета путей сообщения, г. Екатеринбург. E-mail: **marina.vedunova.13.99@gmail.com**

ВЫСОЦКАЯ Виктория Владимировна — аспирантка факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова, специалист-исследователь лаборатории криптографии НПК «Криптонит», г. Москва.
E-mail: **vysotskaya.victory@gmail.com**

ГЕУТ Кристина Леонидовна — старший преподаватель кафедры естественнонаучных дисциплин Уральского государственного университета путей сообщения, г. Екатеринбург.
E-mail: **geutkrl@yandex.ru**

ГРИБАНОВА Ирина Александровна — младший научный сотрудник Института динамики систем и теории управления им. В. М. Матросова СО РАН, г. Иркутск.
E-mail: **the42dimension@gmail.com**

ДОРОНИН Артемий Евгеньевич — студент Новосибирского государственного университета, г. Новосибирск. E-mail: **artem96dor@gmail.com**

ЕГОРУШКИН Олег Игоревич — кандидат физико-математических наук, доцент Сибирского государственного университета науки и технологий имени академика М. Ф. Решетнёва, г. Красноярск.
E-mail: **olegegoruschkin@yandex.ru**

ЕЛИСЕЕВ Владимир Леонидович — кандидат технических наук, руководитель Центра научных исследований и перспективных разработок ОАО «ИнфоТеКС», доцент кафедры управления и интеллектуальных технологий Национального исследовательского университета «Московский энергетический институт», г. Москва. E-mail: **vlad-eliseev@mail.ru**

ЖАРКОВА Анастасия Владимировна — кандидат физико-математических наук, доцент кафедры теоретических основ компьютерной безопасности и криптографии Саратовского национального исследовательского государственного университета им. Н. Г. Чернышевского, г. Саратов.
E-mail: **ZharkovaAV3@gmail.com**

ЗАПОЛЬСКИЙ Максим Михайлович — студент Новосибирского государственного университета, г. Новосибирск. E-mail: **m.zapolskii@g.nsu.ru**

ЗЮБИНА Дарья Александровна — младший научный сотрудник Института математики им. С. Л. Соболева СО РАН, студентка факультета информационных технологий НГУ, исследователь лаборатории криптографии JetBrains Research, г. Новосибирск. E-mail: **d.zyubina@g.nsu.ru**

ИГНАТОВА Анастасия Олеговна — студентка Уральского государственного университета путей сообщения, г. Екатеринбург. E-mail: **anastasiaignatova101@gmail.com**

ИДРИСОВА Валерия Александровна — научный сотрудник Института математики им. С. Л. Соболева СО РАН, г. Новосибирск. E-mail: **vvitkup@yandex.ru**

КАЛГИН Константин Викторович — кандидат физико-математических наук, научный сотрудник Института вычислительной математики и математической геофизики СО РАН, младший научный сотрудник Института математики им. С. Л. Соболева СО РАН, старший преподаватель кафедры параллельных вычислений ФИТ Новосибирского государственного университета, г. Новосибирск. E-mail: **kalginkv@gmail.com**

КИРШАНОВА Елена Алексеевна — Ph. D., доцент Балтийского федерального университета им. И. Канта, г. Калининград. E-mail: **elenakirshanova@gmail.com**

КИШКАН Владимир Владимирович — аспирант кафедры прикладной математики Сибирского государственного университета науки и технологий имени академика М. Ф. Решетнёва, г. Красноярск. E-mail: **kishkan@mail.ru**

КЛИМЕНКО Константин Александрович — директор по продуктам Лаборатории блокчейн, Сбербанк России, г. Москва. E-mail: **blockchain-research@sberbank.ru**

КОЛБАСИНА Ирина Валерьевна — ассистент Сибирского государственного университета науки и технологий имени академика М. Ф. Решетнёва, г. Красноярск. E-mail: **kabaskina@yandex.ru**

КОЛЕСНИКОВ Никита Сергеевич — аспирант Балтийского федерального университета им. И. Канта, г. Калининград. E-mail: **nikolesnikov1@kantiana.ru**

КОНДЫРЕВ Дмитрий Олегович — аспирант факультета информационных технологий Новосибирского государственного университета, исследователь Лаборатории криптографии JetBrains Research, младший научный сотрудник Института математики им. С. Л. Соболева СО РАН, г. Новосибирск. E-mail: **dkondyrev@gmail.com**

КОРЕНЕВА Алиса Михайловна — кандидат физико-математических наук, начальник отдела ООО «Код Безопасности», г. Москва. E-mail: **a.koreneva@securitycode.ru**

КОСОЛАПОВ Юрий Владимирович — кандидат технических наук, доцент Южного федерального университета, г. Ростов-на-Дону. E-mail: **itaim@mail.ru**

КУЗНЕЦОВ Александр Алексеевич — доктор физико-математических наук, профессор, директор института Сибирского государственного университета науки и технологий имени академика М. Ф. Решетнёва, г. Красноярск. E-mail: **alex_kuznetsov80@mail.ru**

КУЦЕНКО Александр Владимирович — аспирант механико-математического факультета Новосибирского национального исследовательского государственного университета, Институт математики им. С. Л. Соболева СО РАН, г. Новосибирск. E-mail: **AlexandrKutsenko@bk.ru**

КЯЖИН Сергей Николаевич — кандидат физико-математических наук, руководитель проектов Лаборатории блокчейн, Сбербанк России, г. Москва. E-mail: **blockchain-research@sberbank.ru**

ЛИПАТОВА Екатерина Сергеевна — студентка Национального исследовательского Томского государственного университета, г. Томск. E-mail: **katrinelipatova@gmail.com**

МАКСИМЛЮК Юлия Павловна — младший научный сотрудник Института математики им. С. Л. Соболева СО РАН, студентка Новосибирского государственного университета, исследователь Лаборатории криптографии JetBrains Research, г. Новосибирск. E-mail: **yumaximlyuk@gmail.com**

МАЛКОВА Ксения Максимовна — аспирантка Владимирского государственного университета, г. Владимир. E-mail: **malkova-xeni@yandex.ru**

МАЛЫГИНА Екатерина Сергеевна — кандидат физико-математических наук, доцент Балтийского федерального университета им. И. Канта, г. Калининград. E-mail: **emalygina@kantiana.ru**

МЕДВЕДЕВ Никита Владимирович — кандидат технических наук, доцент кафедры информационных технологий и защиты информации Уральского государственного университета путей сообщения, г. Екатеринбург. E-mail: **itcrypt@gmail.com**

МЕДВЕДЕВА Наталья Валерьевна — кандидат физико-математических наук, доцент, доцент кафедры Уральского государственного университета путей сообщения, г. Екатеринбург. E-mail: **medvedeva_n_v@mail.ru**

НАБИЕВ Тимур Русланович — студент МГТУ им. Н. Э. Баумана, программист ООО «Код Безопасности», г. Москва. E-mail: **t.nabiev@securitycode.ru**

НОВОСЕЛОВ Семен Александрович — ассистент Балтийского федерального университета им. И. Канта, г. Калининград. E-mail: **snovoselov@kantiana.ru**

ОЛЕФИРЕНКО Денис Олегович — аспирант Балтийского федерального университета им. И. Канта, г. Калининград. E-mail: **denis_cooler_1@mail.ru**

ПАНФЕРОВ Матвей Андреевич — магистрант Новосибирского государственного университета, исследователь Лаборатории криптографии JetBrains Research, г. Новосибирск. E-mail: **m.panferov@g.nsu.ru**

ПЕРОВ Артём Андреевич — старший преподаватель Новосибирского государственного университета экономики и управления (НИНХ), г. Новосибирск. E-mail: **perov_artem@inbox.ru**

ПЕСТУНОВ Андрей Игоревич — кандидат физико-математических наук, доцент, заведующий кафедрой информационных технологий Новосибирского государственного университета экономики и управления (НИНХ), г. Новосибирск. E-mail: **pestunov@gmail.com**

ПИНТУС Георгий Михайлович — студент Новосибирского государственного университета, г. Новосибирск. E-mail: **g.pintus@g.nsu.ru**

РОМАНЬКОВ Виталий Анатольевич — доктор физико-математических наук, профессор, заведующий кафедрой Омского государственного университета им. Ф. М. Достоевского, главный научный сотрудник Института математики им. С. Л. Соболева СО РАН (Омский филиал), г. Омск. E-mail: **romankov48@mail.ru**

РЫБАЛОВ Александр Николаевич — кандидат физико-математических наук, старший научный сотрудник лаборатории комбинаторных и вычислительных методов алгебры и логики Института математики им. С. Л. Соболева СО РАН, г. Омск. E-mail: **alexander.rybalov@gmail.com**

САФОНОВ Константин Владимирович — доктор физико-математических наук, профессор, заведующий кафедрой Сибирского государственного университета науки и технологий имени академика М. Ф. Решетнёва, г. Красноярск. E-mail: **safonovkv@rambler.ru**

СЕМЁНОВ Александр Анатольевич — кандидат технических наук, доцент, заведующий лабораторией Института динамики систем и теории управления им. В. М. Матросова СО РАН, г. Иркутск. E-mail: **biclop.rambler@yandex.ru**

СОФРОНОВА Дарья Алексеевна — студентка Новосибирского государственного университета, исследователь Лаборатории криптографии JetBrains Research, г. Новосибирск. E-mail: **d.sofronova1@g.nsu.ru**

СУТОРМИН Иван Александрович — младший научный сотрудник Института математики им. С. Л. Соболева СО РАН, студент Новосибирского государственного университета, г. Новосибирск. E-mail: **ivan.sutormin@gmail.com**

ТЕРЕБИН Богдан Андреевич — студент Саратовского национального исследовательского государственного университета им. Н. Г. Чернышевского, г. Саратов. E-mail: **bogdan.terebin@yandex.ru**

ТИТОВ Сергей Сергеевич — доктор физико-математических наук, профессор Уральского государственного университета путей сообщения, г. Екатеринбург. E-mail: **sergey.titov@usaaa.ru**

ТОКАРЕВА Наталья Николаевна — кандидат физико-математических наук, старший научный сотрудник Института математики им. С. Л. Соболева СО РАН, доцент Новосибирского государственного университета, г. Новосибирск. E-mail: **tokareva@math.nsc.ru**

ТУРЧЕНКО Олег Юрьевич — аспирант Южного федерального университета, г. Ростов-на-Дону. E-mail: **olegmmcs@gmail.com**

ФОМИЧЁВ Владимир Михайлович — доктор физико-математических наук, профессор, научный консультант ООО «Код Безопасности», профессор Финансового университета при Правительстве РФ, профессор НИЯУ МИФИ, ведущий научный сотрудник ФИЦ ИУ РАН, г. Москва.
E-mail: **fomichev.2016@yandex.ru**

ЦАРЕГОРОДЦЕВ Кирилл Денисович — аспирант МГУ им. М. В. Ломоносова, специалист-исследователь НПК «Криптонит», г. Москва. E-mail: **kirill94_12@mail.ru**

ЧЕРЕДНИК Игорь Владимирович — преподаватель РТУ МИРЭА, г. Москва.
E-mail: **p.n.v.k.s@mail.ru**

ЧЕРЕМУШКИН Александр Васильевич — доктор физико-математических наук, профессор, член-корреспондент Академии криптографии РФ, г. Москва. E-mail: **avc238@mail.ru**

ШАПОРЕНКО Александр Сергеевич — младший научный сотрудник Института математики им. С. Л. Соболева СО РАН, студент Новосибирского государственного университета, исследователь Лаборатории криптографии JetBrains Research, г. Новосибирск.
E-mail: **shaporenko.alexandr@gmail.com**

АННОТАЦИИ ДОКЛАДОВ НА АНГЛИЙСКОМ ЯЗЫКЕ

SECTION 1

Vedunova M., Geut K., Ignatova A., Titov S. **REFRACTIVE BIJECTIONS IN STEINER TRIPLES.** The paper deals with refractive bijections in Steiner triples used in the construction of matroids and secret sharing schemes. Refractors are understood to mean mappings F of a quasigroup into itself satisfying the condition $F(x*y) \neq F(x)*F(y)$ for any $x \neq y$. The necessary conditions for the existence of APN-bijections in $\mathrm{GF}(2^n)$ are found, for $N = 7$ the superposition of any two refractive bijections is not refractive. It is found that for $N = 9$, 13 and $2^n - 1$ elements for odd n not divisible by three, there are three Steiner triples systems without common triples. Refractive bijections are proposed for systems of Steiner triples without common triples for $N = 13$. A counterexample is obtained to the hypothesis that each homogeneous matroid defines a certain block scheme using sets of refractive bijections, for $N = 7$ such S, S', S'' do not exist. Functions that are APN-bijections are given. The condition allowing to construct homogeneous matroids that are not reduced to block scheme used in secret sharing schemes using Steiner linear triples systems is revealed, and a refractive bijection that is not an APN-function is also found, for instance $F(x) = x^{-3}$.
Keywords: *refracting bijections, Steiner quasigroups, matroids.*

Medvedev N. V., Titov S. S. **ON HOMOGENEOUS MATROIDS CORRESPONDING TO BLOCK-SCHEMES.** The paper deals with relationship of homogeneous matroids and block-schemes. This problem is related to the study of access structures of ideal perfect secrets sharing schemes. By homogeneous matroids we mean an equal degree of cycles, where, perhaps, not all subsets of this degree are cycles. If power of cycles is equal to five, then it is proved that homogeneous connected separating matroid will be uniform. However, if the matroid is connected and separating, then the dual matroid will be simple. It is proved that if each cycle of homogeneous separating connected matroid is a hyperplane, then a block-scheme corresponds to it.
Keywords: *homogeneous matroids, secret sharing schemes, block-schemes, cycles.*

Olefirenko D. O., Kirshanova E. A., Malygina E. S., Novoselov S. A. **AN ALGORITHM FOR COMPUTING THE STICKELBERGER ELEMENTS FOR IMAGINARY MULTIQUADRATIC FIELDS.** In this paper we present an algorithm for computing the Stickelberger ideal for multiquadratic fields $K = \mathbb{Q}(\sqrt{d_1}, \sqrt{d_2}, \ldots, \sqrt{d_n})$, where $d_i \equiv 1 \pmod 4$ for $i = 1, \ldots, n$ and d_i's are pair-wise co-prime. Our result is based on the work of R. Kucera [J. Number Theory 56, 1996]. We systematize the ideas of this work, put them into explicit algorithms, prove their correctness and complexity. For $2^n = [K : \mathbb{Q}]$, our algorithm runs for time $\widetilde{\mathcal{O}}(2^n)$. We hope that the obtained results will serve as the first step towards solving the shortest vector problem for ideals of multiquadratic fields, which is the core problem in lattice-based cryptography.
Keywords: *multiquadratic number field, Stickelberger ideal, Stickelberger element, the shortest vector problem.*

SECTION 2

Agievich S. V. **ON THE CONTINUATION TO BENT FUNCTIONS AND UPPER BOUNDS ON THEIR NUMBER.** A Boolean bent function f of n variables is a continuation of a Boolean function g of $k < n$ variables if g is a restriction of f to a fixed affine plane of dimension k. We prove that a continuation always exists if $k \leqslant n/2$. We obtain an upper bound for the number of continuations. The bound is strengthened in the case $k = n - 1$, when g is a near-bent function. As a result, we improve the known upper bounds for the number of bent functions. More precisely, we show that for even $n \geqslant 6$ there are no more than

$$c_n\, 2^{2^{n-2}-n/2+5/2} \left(\frac{B(n/2, n-1) - B(n/2-1, n-1)}{2^{2^{n/2}-n/2-1}} + B(n/2-1, n-1) \right)$$

bent functions of n variables. Here $c_n = \exp(-1/2 + 23/(18 \cdot 2^{n-2}))/\sqrt{\pi}$ and $B(d, n) = 2^{\binom{n}{0}+\binom{n}{1}+\ldots+\binom{n}{d}}$.

Keywords: *bent function, number of bent functions, near-bent function, affine plane.*

Kutsenko A. V. **ON METRICAL PROPERTIES OF THE SET OF SELF-DUAL BENT FUNCTIONS.** For every bent function f its dual bent function $\widetilde{f}$ is uniquely defined. If $\tilde{f} = f$ then f is called *self-dual bent* and it is called *anti-self-dual bent* if $\tilde{f} = f \oplus 1$. In this work we give a review of metrical properties of the set of self-dual bent functions. We give a complete Hamming distance spectrum between self-dual Maiorana — McFarland bent functions. The set of Boolean functions which are maximally distant from the set of self-dual bent functions is discussed. We give a characterization of automorphim groups of the sets of self-dual and anti-self-dual bent functions in n variables as well as the description of isometric mappings that define bijections between the sets of self-dual and anti-self dual bent functions. The set of isometric mappings which preserve the Rayleigh quotient of a Boolean function is given. As a corollary all isometric mappings which preserve bentness and the Hamming distance between bent function and its dual are given.

Keywords: *Boolean function, self-dual bent function, Hamming distance, isometric mapping, metrical regularity, automorphism group, Rayleigh quotient of Sylvester Hadamard matrix.*

Lipatova E. S. **CRYPTOGRAPHIC PROPERTIES OF SOME VECTORIAL BOOLEAN FUNCTIONS COMPOSITIONS.** Three classes of vectorial Boolean functions are considered such that each of their coordinate functions essentially depends on a given number of variables. The experimental results for the cryptographic properties (algebraic degree, algebraic immunity, nonlinearity, differential uniformity) of compositions of functions from these classes are presented.

Keywords: *vectorial Boolean functions, nonlinearity, algebraic immunity, differential uniformity.*

Maksimluk J. P. **CRYPTOGRAPHIC PROPERTIES OF ORTHOMORPHIC PERMUTATIONS.** In this paper, we consider bijective mappings $F : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ called orthomorphisms such that the mappings $G(x) = F(x) \oplus x$ are also bijective. It is used in the Lai — Massey scheme as a mixing element between rounds and it also can be used to construct cryptographically strong S-boxes. The main cryptographic properties are studied, namely nonlinearity and differential uniformity. It was revealed that, for $n = 2, 3, 4$, the linear approximation tables of orthomorphisms consist of the values 0 and $\pm 2^{n-1}$, and

the difference distribution tables consist of the values 0 and 2^n. It turned out that orthomorphisms of a small number of variables are not resistant to linear and differential cryptanalysis.
Keywords: *orthomorphic permutation, linear approximation table, difference distribution table.*

Pintus G. M. **ON THE DECOMPOSITION OF A VECTORIAL BOOLEAN FUNCTION INTO A COMPOSITION OF TWO FUNCTIONS.** In the paper, we prove that if a vectorial Boolean function F in n variables, $\deg(F) = d > 2$, is decomposable, then the function $F' = A_2 \circ F \circ A_1$, where A_1, A_2 are arbitrary affine (n, n)-permutations, is also decomposable; and if $F(x) = G(H(x))$, $\max\{\deg(F), \deg(H)\} = d' < d$, function H is invertible and $\deg(H^{-1}) \leqslant d'$, then the function $F'' = F + A_0$ is decomposable for any affine function A_0. The construction of a decomposable vectorial Boolean function of the third degree in an arbitrary number of variables is presented. A computational experiment showed that all vectorial Boolean functions of the third degree in three variables are decomposable.
Keywords: *vectorial Boolean function, decomposition, threshold implementation.*

Sutormin I.A. **AN ESTIMATION OF THE NONLINEARITY OF BALANCED BOOLEAN FUNCTIONS GENERATED BY GENERALIZED DOBBERTIN'S CONSTRUCTION.** A generalization of the Dobbertin's construction for highly nonlinear balanced Boolean functions is proposed. The Walsh — Hadamard spectrum is studied and estimates of the spectral radius of the proposed functions are obtained. An exact upper bound for the spectral radius (lower bound for nonlinearity) is proved, and a method for constructing a balanced function Θ in $2n$ variables using a balanced θ in $n - k$ variables with spectral radius $R_\Theta = 2^n + 2^k R_\theta$ is proposed. Here, R_Θ and R_θ are the spectral radii of Θ and θ respectively.
Keywords: *boolean functions, bent functions, balancedness, nonlinearity, spectral radius.*

Shaporenko A. S. **CONNECTIONS BETWEEN QUATERNARY AND COMPONENT BOOLEAN BENT FUNCTIONS.** This paper is about quaternary bent functions. Function $g : \mathbb{Z}_4^n \to \mathbb{Z}_4$ is called quaternary in n variables. It was proven that bentness of a quaternary function $g(x + 2y) = a(x, y) + 2b(x, y)$ doesn't directly depend on the bentness of Boolean functions b and $a \oplus b$. The number of quaternary bent functions in one and two variables is obtained with a description of properties of Boolean functions b and $a \oplus b$. Two simple constructions of quaternary bent functions in any number of variables are presented. The first one is given by the formula $g(x_1+2x_{n+1}, \ldots, x_n+2x_{2n}) = \sum_{i=1}^{n} 2x_i x_{i+n} + cx_j$, $c \in \mathbb{Z}_2$ and $j \in \{1, \ldots, n\}$. The second construction allows one to get a bent function $g'(x + 2y) = 3a(x, y) + 2b(x, y)$, where $g(x + 2y) = a(x, y) + 2b(x, y)$ is bent.
Keywords: *quaternary functions, Boolean functions, bent function.*

Kalgin K. V., Idrisova V. A. **ON A SECONDARY CONSTRUCTION OF QUADRATIC APN FUNCTIONS.** Almost perfect nonlinear functions possess the optimal resistance to the differential cryptanalysis and are widely studied. Most known constructions of APN functions are obtained as functions over finite fields $\mathbb{F}_{2^n}$ and very little is known about combinatorial constructions in $\mathbb{F}_2^n$. We consider how to obtain a quadratic APN function in $n + 1$ variables from a given quadratic APN function in n variables using special restrictions on new terms.

Keywords: *vectorial Boolean function, APN function, quadratic function, secondary construction.*

Zapolskiy M. M., Tokareva N. N. **ON ONE-TO-ONE PROPERTY OF A VECTORIAL BOOLEAN FUNCTION OF THE SPECIAL TYPE.** S-boxes are widely used in cryptography. In particular, they form important components of SP and Feistel networks. Mathematically, S-box is a vectorial Boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ that should satisfy several cryptographic properties. Usually $n = m$. We study one-to-one property of a vectorial Boolean function constructed in a special way on the base of a Boolean function and a permutation on n elements. The number of all one-to-one functions of this type is calculated.

Keywords: *Boolean function, vectorial Boolean function, S-box.*

Zyubina D. A., Tokareva N. N. **CRYPTOGRAPHIC PROPERTIES OF A SIMPLE S-BOX CONSTRUCTION BASED ON A BOOLEAN FUNCTION AND A PERMUTATION.** We propose a simple method of constructing S-boxes using Boolean functions and permutations. Let π be an arbitrary permutation on n elements, f be a Boolean function in n variables. Define a vectorial Boolean function $F_\pi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ as $F_\pi(x) = (f(x), f(\pi(x)), f(\pi^2(x)), \ldots, f(\pi^{n-1}(x)))$. We study cryptographic properties of F_π such as high nonlinearity, balancedness, low differential δ-uniformity in dependence on properties of f and π for small n.

Keywords: *Boolean function, vectorial Boolean function, S-box, high nonlinearity, balancedness, low differential δ-uniformity, high algebraic degree.*

SECTION 3

Kirshanova E. A., Kolesnikov N. S, Malygina E. S., Novoselov S. A. **POST-QUANTUM SIGNATURE PROPOSAL FOR STANDARDISATION.** In this paper, we propose an algebraic lattice-based signature scheme. The design of the proposal follows the Fiat — Shamir paradigm. Our scheme is proved secure in the quantum random oracle model and achieves security against $\mathsf{UF-sCMA}$ adversaries. The concrete parameters to instantiate the scheme that achieves 100 bits of security are proposed. Thanks to the algebraic structure of the construction, the scheme is flexible in security levels so that we can achieve trade-offs between speed and security. Our proposal may serve as the basis for a standard of lattice-based schemes.

Keywords: *signature scheme, lattice-based cryptography, post-quantum cryptography, Fiat — Shamir transform.*

Medvedeva N. V., Titov S. S. **CONSTRUCTIONS OF NON-ENDOMORPHIC PERFECT CIPHERS.** This work is dealing with constructions of Shannon perfect ciphers (which are absolutely immune against the attack on ciphertext, according to Shannon). Based on the equivalence relation on the set of keys, sufficient conditions are obtained for that the encoding tables of non-endomorphic (endomorphic) perfect ciphers do not contain Latin rectangles (squares). Key equivalence refers to the following: two different keys are equivalent in cipher-value x_i if the cipher-value x_i on these keys is encrypted to the same code designation. In this case, pairwise different keys $k_1, k_2, k_3, \ldots, k_{n-1}, k_n$ form a cycle of length n if there is such a sequence of cipher-values that: 1) the neighboring cipher-values are different; 2) the keys $k_1, k_2, k_3, \ldots, k_{n-1}, k_n, k_1$ are sequentially equivalent in the corresponding cipher-values. If n is an odd number, then the keys $k_1, k_2, \ldots, k_n$ form an odd-length cycle. It is proved that if the keys $k_1, k_2, \ldots, k_n$ form an odd-length cycle, then

this encoding table does not contain Latin rectangles. Example of such constructions is given.

Keywords: *perfect ciphers, endomorphic ciphers, non-endomorphic ciphers.*

Perov A. A., Pestunov A. I. **CONSTRUCTION OF DISTINGUISHERS FOR ITERATIVE BLOCK CIPHERS ON THE BASE OF NEURAL NETWORKS.** We present a new machine learning approach for creating distinguishers (or distinguishing attacks) for iterative block ciphers with a variable number of rounds. A main idea of the approach is based, firstly, on the observation that block cipher ciphertexts (if represented as images) have different textures (patterns) depending on the number of rounds and, secondly, on the fact that modern neural networks can classify images with high accuracy. So, we suggest to represent ciphertexts as images and train neural network to recognize them. In such a case, a level of the ciphertexts randomness can be determined by a classification error value. We introduce two methods that can be used for practice: the "etalon-ciphertext" method and the "neighbor-rounds" method. It has been experimentally demonstrated that the "etalon" method is slightly more accurate than the "neighbor-rounds" method, however, it requires an etalon sequence of true-random numbers or, at least, numbers with good statistical properties. At the same time, the second method requires only one cipher. In our experiments, we used the AES256 cipertext as the etalon, however, any other proper random number generator can be used. In both cases, the classification error converges to 50 % when the number of rounds increases. Both methods can also be employed in order to estimate the minimal number of rounds which provides acceptable statistical properties of the ciphertext.

Keywords: *block cipher, machine learning, neural network, statistical analysis, distinguishing attack.*

Roman'kov V. A. **ABOUT THE HIDDEN COMPACT WAY TO STORE DATA.** A fundamentally new method for compact data storage in a hidden form in a "magic" square using discrete sequence integrals is proposed. Each of the data can be extracted in a uniform way. A comparison is made with other possible methods of such storage.

Keywords: *data, storage, hidden, compactness, access.*

Fomichev V. M., Bobrovskiy D. A., Koreneva A. M. **EXPERIMENTAL ESTIMATES OF THE COMPUTATIONAL COMPLEXITY OF ONE CLASS OF CRYPTOALGORITHMS BASED ON THE GENERALIZATION OF FEISTEL NETWORKS.** The development of information technologies and the need to protect information indicate the relevance of developing new cryptographic algorithms, such as block ciphers with different block sizes that correspond to modern requirements for cryptographic stability and performance. This paper presents the results of experimental studies of algorithm 256-3 performance (block size is 256 bits), proposed by Russian researchers in 2018. This paper provides a performance comparison between 256-3 and well-known block ciphers. The comparison has been conducted by running implementations of algorithms in C++ programming language using Crypto++ library. The results showed that the 256-3 algorithm runs around 24.57 cycles per byte and performance of 256-3 from 1.2 to 2.6 times higher than the performance of the algorithms Magma (GOST 28147-89), Kuznyechik (GOST 34.12-2018), SEED, HIGHT, Camellia-256, Kalyna-256/256, MARS-256, CAST-256, which indicates that 256-3 is a positive (from the synthesis position).

Keywords: *block cipher performance, block ciphers benchmarks, 256-3, GOST, Magma, Kuznyechik, AES, Rijndael, SEED, SM4, HIGHT, Camellia, Kalyna, MARS, CAST, RC6, Crypto++.*

Fomichev V. M., Koreneva A. M., Nabiev T. R. **CHARACTERISTICS OF THE DATA INTEGRITY CHECK ALGORITHM BASED ON ADDITIVE GENERATORS AND *S*-BOXES.** During software analysis, the integrity control of large data arrays is relevant. In solving this task it is important to provide an acceptable compromise between cryptographic properties of the integrity check algorithm and the resources necessary for its implementation. We propose the algorithm for generation of 128-bit integrity check value (ICV) for data blocks of size 1 KB (1024 bytes). This algorithm provides positive (from the synthesis position) operational and cryptographic properties and uses the transformations of additive generators and s-boxes. The algorithm is implemented by the function $\psi(g^t)\colon V_{2^{13}} \to V_{128}$ with the full mixing of the input data. For $6 \leqslant t \leqslant 100$, each bit of the ICV essentially depends on all the bits of the input block. If you randomly choose the initial state u, the probability of obtaining the corresponding ICV code Q is estimated by 2^{-128}. The average number of the tested pairs of blocks (u, u'), where $u \neq u'$ and $Q(u) = Q(u')$, is approximately equal to 2^{64}. The computational complexity of the function $\psi(g^t)$ is in the order of $t(5u + 8v)$, where u is the computational complexity of adding two numbers modulo 2^{64}, and v is the computational complexity of the s-box calculation. According to the conducted experiments, the speed of ICV generation varies from 3500 (t=6) to 250 Mbit/s (t=96), respectively. At the same values of t, the time of ICV generation varies from 18 to 250 μs.

Keywords: *additive generators, data integrity control, matrix-graph approach, mixing properties, shift registers.*

Tsaregorodtsev K. D. **ANALYSIS OF BLOCK CIPHER MODES OF OPERATION FOR RFID DEVICES.** The RFID system consists of radio-tag and interrogator. The main purpose of devices is authentication with a transmission of small amount of data between tag and interrogator. We study modes that require only block encryption to be implemented in the tag (due to hardware limitations). CTR, OFB and modified CBC modes were analyzed. Modified CBC mode CBC[F] can be specified as follows:

$$C_0 = \mathrm{IV}, \quad C_i = F_k(C_{i-1} \oplus M_i),$$

where F_k denotes either encryption ($F_k = E_k$) or decryption ($F_k = E_k^{-1}$) of one block of data on the key k, M_i is the i-th block of plaintext, C_i is the i-th block of ciphertext, IV is an initialization vector. Assume that the length of the key is k bits, the length of the one block of data is n bits, and consider an adversary that runs time no more than t, makes no more than q oracle queries, each of length no more than m blocks. Then, given an oracle access to the CBC[E_k] and $\widehat{\mathrm{CBC}} = \mathrm{CBC}[E_k^{-1}]$ modes, the maximal advantage in the LOR-experiment can be bounded from above by:

$$\mathrm{Adv}^{\mathrm{LOR}}_{\mathrm{CBC},\widehat{\mathrm{CBC}}}(t, q, m) \leqslant \frac{3q^2m^2}{2^n - qm} + \frac{t+q}{2^{k-1}}.$$

Keywords: *provable security, RFID, mode of operation.*

Cherednik I. V. **ONE APPROACH TO CONSTRUCTING A MULTIPLY TRANSITIVE CLASS OF BLOCK TRANSFORMATIONS.** Let Ω be an arbitrary finite set, $\mathcal{B}(\Omega)$ — the collection of all binary operations defined on the set Ω, $\mathcal{B}^*(\Omega)$ — the family of all binary operations that are invertible in the right variable, $x_1, \ldots, x_n$ — variables over Ω, and $*_1, \ldots, *_k$ — general symbols of binary operations. A fixed cortege

$W = (w_1, \ldots, w_m)$ of formulas in the alphabet $\{x_1, \ldots, x_n, *_1, \ldots, *_k\}$ implements the mapping $W^{F_1,\ldots,F_k} \colon \Omega^n \to \Omega^m$ when replacing symbols $*_1, \ldots, *_k$ with an arbitrary binary operations $F_1, \ldots, F_k \in \mathcal{B}(\Omega)$, respectively. In this paper we offer a visual representation of the transformation family $\{W^{F_1,\ldots,F_k} : F_1, \ldots, F_k \in \mathcal{B}^*(\Omega)\}$ in the form of a binary functional network. This representation allows us to strictly describe the methods of research on the multiply transitivity of an arbitrary family $\{W^{F_1,\ldots,F_k} : F_1, \ldots, F_k \in \mathcal{B}^*(\Omega)\}$. In addition, network view makes it possible to construct cortege of formulas $W = (w_1, \ldots, w_n)$ such that the family $\{W^{F_1,\ldots,F_k} : F_1, \ldots, F_k \in \mathcal{B}^*(\Omega)\}$ is multiply transitive. Moreover, some block ciphers (Blowfish, Twofish, etc), in which the S-boxes depend on the key, can be "approximated" by family of the form $\{W^{F_1,\ldots,F_k} : F_1, \ldots, F_k \in \mathcal{B}^*(\Omega)\}$ and, as a result, it becomes possible to evaluate the multiple transitivity of such ciphers.

Keywords: *block transformation, multiply transitive class of block transformations, functional binary network.*

Cheremushkin A. V. **ELABORATION OF SELFISH-MINE STRATEGY.** As it was shown by Ittay Eyal and Emin Gün Sirer, the Bitcoin mining protocol is not incentive-compatible, because there exists an attack in which colluding miners obtain a revenue larger than their fair share. We describe an elaboration of Selfish-Mine Strategy and present an extended model of selfish mining based on independency hypothesis: both groups are made their work independently from each other. We describe a new state machine modelling selfish pool strategy. Let the selfish pool has mining power of p, $0 < p < 1/2$, and the others of $(1 - p)$. We also consider the situation in which the others mine a block on the previously private branch (frequency $\gamma(1 - p)$), and the others mine a block on the public branch (frequency $(1 - \gamma)(1 - p)$). Main result is an elaboration of an interval, in which selfish miners will earn more than their relative mining power: 1) for a given p, a pool of size p obtains a revenue larger than its relative size for p in the following range: $0 < p \leqslant 0.429$ (the left bound coinsides with $\gamma = 1$, and the right bound coinsides with $\gamma = 0$); 2) for a given p, a pool of size p obtains a revenue larger than a revenue of other group in the following range: $0.358 \leqslant p \leqslant 0.449$.

Keywords: *blockchain, mining, Markov model, state machine.*

Bonich T. A., Panferov M. A., Tokareva N. N. **ON THE NUMBER OF UNSUITABLE BOOLEAN FUNCTIONS IN CONSTRUCTIONS OF FILTER AND COMBINING MODELS OF STREAM CIPHERS.** It is well known that every stream cipher is based on a good pseudorandom generator. For cryptographic purposes, we are interested in generation of pseudorandom sequences of the maximal possible period. A feedback register is one of the most known cryptographic primitives that is used in construction of stream generators. We analyze periodic properties of pseudorandom sequences produced by filter and combiner generators equipped with nonlinear Boolean functions. We determine which nonlinear functions in these schemes lead to pseudorandom sequences of not maximal possible period. We call such functions unsuitable and count the exact number of them for an arbitrary n.

Keywords: *stream cipher, filter generator, combiner generator, gamma, Boolean function.*

Kosolapov Y. V., Turchenko O. Y. **EFFICIENT S-REPETITION METHOD FOR CONSTRUCTING AN IND-CCA2 SECURE MCELIECE MODIFICATION IN THE STANDARD MODEL.** The paper deals with the construction of IND-CCA2-secure modification of the McEliece cryptosystem in the standard model. The modification uses S-repetition encryption of $S/2$ various messages with one common secret permutation, in contrast to other modifications that use S-repetition encryption of one message. Thus,

this modification provides IND-CCA2-security with an efficient information transfer rate.
Keywords: *post-quantum cryptography, McEliece-type cryptosystem,* IND-CCA2*-security, S-repetition encryption.*

SECTION 4

Eliseev V. L. **NEURAL NETWORK OBFUSCATION FOR COMPUTATIONS OVER ENCRYPTED DATA.** An approach to neural network cryptographic obfuscation of computations is proposed. Applying the previously obtained results on the property of strict obfuscation of indistinguishability for a neural network approximator, we propose to use neural networks to perform arithmetic and other operations on encrypted data, thus realizing the idea of using homomorphic encryption to perform trusted computations in an untrusted environment. The cryptographic properties of this mechanism are evaluated and compared with traditional approaches to encryption based on the secret key. The advantages and disadvantages of neural networks in relation to the problem of obfuscation and processing of encrypted data are discussed.
Keywords: *artificial neural network, obfuscation, homomorphic encryption, secrecy estimation.*

Kondyrev D. O. **METHOD FOR HIDING PRIVATE DATA IN THE BLOCKCHAIN TENDER SYSTEM.** A new method has been proposed to solve the problem of information privacy in open blockchain systems using the zk-SNARK cryptographic zero-knowledge proof protocol. The proposed method has been implemented as a cryptographic scheme based on the libsnark library. To integrate the cryptographic scheme into the system, the Ethereum C++ client has been modified, where new functions and an interface for working with them in the form of precompiled contracts has been added.
Keywords: *tenders, distributed systems, blockchain, zero-knowledge proof, zk-SNARK, Ethereum platform.*

Kyazhin S. N., Klimenko K. A. **VALIDATION-FREE OFFCHAIN TRANSACTIONS WITH UNLINKABLE DOUBLE SPEND DETECTION.** The so-called layer-two protocols are a class of blockchain scaling solutions. They allow to minimize onchain traffic, and therefore make state transitions (payments, for example) faster and more suitable for everyday use, while still preventing double spend attacks. Unfortunately, these solutions also have some downsides and tradeoffs (channel capacity, route availability, operator availability, etc.). In this work we study the possibility of simplifying and improving existing protocols for offchain transactions and describe a scheme that, without transaction validation, allows to detect a double spender and not trace other transactions. This scheme is based on the anonymous transferable e-cash system. We use an offchain analogue of the UTXO model, therefore there are offchain transactions for issue, transfer and redeem of a so-called note, containing a number that can be used as a secret key to make the corresponding token transfer transaction onchain.
Keywords: *blockchain, offchain, unlinkability, double spend detection.*

SECTION 5

Vysotskaya V. V. **NEW ESTIMATES FOR DIMENSION OF REED — MULLER SUBCODES WITH MAXIMUM HADAMARD SQUARE.** The existence of some structure in a code can lead to decreasing the security of the whole system built on it. Often subcodes are used to "disguise" the code as a "general-looking" one. However, the security of subcodes with Hadamard square equal to the square of the base code is reduced to the security of the latter. Thus, it is necessary to take this property into account during both synthesis and cryptanalysis of code schemes. In the paper, the authors analyse the minimum number of monomials of degree r, which, when added to the code $RM(r-1,m)$, result in a subcode with maximal Hadamard square, that is, coinciding with $RM(2r,m)$. The problem is reformulated in terms of hypergraphs where vertices correspond to variables and each monomial of degree k is associated with a k-edge. A set of r-edges covering all $2r$-sets is searched. The minimum size of such a set is estimated from below as $w(m,r) \geqslant \sqrt{\gamma + 2\mathrm{C}_m^{2r}} + \sqrt{\gamma}$, where $\gamma = \sum\limits_{i=\max\{1,3r-m\}}^{r-1} \mathrm{C}_r^i$. A greedy algorithm constructing a "good" set is proposed to obtain an upper bound. At each step the algorithm adds a new r-edge choosing the "least covered" vertices.

Keywords: *post-quantum cryptography, code-based cryptography, Reed — Muller subcodes, Reed — Muller codes, Hadamard product, McEliece cryptosystem.*

Zharkova A. V. **ON NUMBER OF INACCESSIBLE STATES IN FINITE DYNAMIC SYSTEMS OF COMPLETE GRAPHS ORIENTATIONS.** Finite dynamic systems of complete graphs orientations are considered. The states of such a system (Γ_{K_n}, α), $n > 1$, are all possible orientations of a given complete graph K_n, and evolutionary function α transforms a given state (tournament) G by reversing all arcs in G that enter into sinks, and there are no other differences between the given G and the next $\alpha(G)$ states. In this paper, formulas for calculating the number of inaccessible and the number of accessible states in finite dynamic systems of complete graphs orientations are given. Namely, in the considered system (Γ_{K_n}, α), $n > 1$, the state $G \in \Gamma_{K_n}$ is inaccessible if and only if in this digraph G there is no source and there is a sink. In the finite dynamic system (Γ_{K_n}, α), $n > 1$, the number of inaccessible states is $n\big(2^{(n-1)(n-2)/2} - (n-1)2^{(n-2)(n-3)/2}\big)$ and the number of accessible states is $2^{n(n-1)/2} - n\big(2^{(n-1)(n-2)/2} - (n-1)2^{(n-2)(n-3)/2}\big)$. The corresponding table is given for the finite dynamic systems of complete graphs orientations with the number of vertices from 2 to 10.

Keywords: *accessible state, complete graph, evolutionary function, finite dynamic system, graph, graph orientation, inaccessible state, index, sink, source, tournament.*

Terebin B. A., Abrosimov M. B. **ON THE OPTIMALITY OF GRAPH IMPLEMENTATIONS WITH PRESCRIBED CONNECTIVITIES.** Connected graphs are of great interest in applications, i.e., in design of reliable systems. The vertex connectivity k of a graph G is the minimum number of vertices whose removal leads to a disconnected or trivial graph. Analogously, the edge connectivity λ of a graph G is the minimum number of edges whose removal leads to a disconnected or trivial graph. They are related with the minimum vertex degree δ by Whitney inequality: $k \leqslant \lambda \leqslant \delta$. G. Chartrand and F. Harary proved that this result is not improving in the sense that for any natural numbers a, b, c, such that $0 < a \leqslant b \leqslant c$, we can construct a graph for which $k = a$, $\lambda = b$, $\delta = c$. In their proof, Chartrand and Harary proposed the graph with the number of vertices $2(c+1)$ and the number of edges $c(c+1) + b$, and the prescribed values of vertex connection, edge

connection, and the minimum degree of vertices. In this paper, we consider the problem of constructing the corresponding implementation with the smallest possible number of vertices and edges. Main results: if $a \leqslant b < c$, then the minimun number of vertices is $2(c+1)$, if $a = b = c$, then it is $c + 1$, and if $a \leqslant b = c$, then the minimum number of vertices is $2(c + 1) - a$.
Keywords: *vertex connectivity, edge connectivity, Whitney's inequality.*

SECTION 6

Egorushkin O. I., Kolbasina I. V., Safonov K. V. **GEOMETRIC CONDITION OF FORMAL GRAMMARS SOLVABILITY.** In this paper, we continue the development of a method for studying formal grammars, which means systems of non-commutative polynomial equations. Such systems are solved in the form of formal power series (FPS) that represent non-terminal alphabet characters through terminal alphabet characters; the first component of the solution is a formal language. The method developed by the authors is based on the study of the commutative image of grammar and formal language. Namely, every FPS is associated with its commutative image, which is obtained if we assume that all symbols are commutative variables. A theorem that gives a sufficient geometric condition for the formal grammar to have a unique solution in the form of FPS is obtained: if the commutative images of non-commutative equations of a system define smooth complex analytical hypersurfaces at the point 0, and the normals to them drawn from this point are linearly independent, then the system of non-commutative equations has a unique solution in the form of FPS.
Keywords: *systems of polynomial equations, non-commutative variables, formal power series, commutative image, analytic hypersurface.*

Kishkan V. V., Safonov K. V. **AN ALGORITHM FOR SOLVING THE EXTENDED PARSING PROBLEM.** The statement of the extended problem of parsing is being clarified: to develop a deadlock algorithm that allows one to establish whether a given monomial can be derived using the system of productions that form the grammar of a context-free programming language, and also describe all the derivations of this monomial, if they exist. The description of the monomial derivation is as follows: to determine which productions from the grammar of the language, how many times and in what order are used to derive this monomial, which is equivalent to the construction of all output trees. The paper proposes an algorithm for solving the extended problem of parsing, based on a hierarchy of marked brackets; labeling of brackets shows what productions they are assigned to, and allows you to trace the order of their use. The complexity of this algorithm is equal to $O(Ng^{d^N})$, where g and d are some integers, however, the algorithm has a simple software implementation.
Keywords: *extended parsing problem, context-free language, complicity of algorithm.*

Rybalov A. N. **ON GENERIC COMPLEXITY OF THE PROBLEM TO REPRESENT NATURAL NUMBERS BY SUM OF TWO SQUARES.** Generic-case approach to algorithmic problems was suggested by Miasnikov, Kapovich, Schupp and Shpilrain in 2003. This approach studies behavior of an algorithm on typical (almost all) inputs and ignores the rest of inputs. In this paper, we study the generic complexity of the problem to represent natural numbers by sum of two squares. This problem, going back to Fermat and Euler, is closely related to the problem of integer factorization and the quadratic residuosity problem modulo composite numbers, for which no efficient algorithms

are known. We prove that under the condition of worst-case hardness and P = BPP, for the problem of representation of natural numbers by sum of two squares there is no polynomial strongly generic algorithm. A strongly generic algorithm solves a problem not on the whole set of inputs, but on a subset, the sequence of frequencies which with increasing size converges exponentially fast to 1.
Keywords: *generic complexity, sums of squares, Diophantine equations.*

SECTION 7

Antonov K. V., Semenov A. A. **APPLICATION OF SAT ORACLES FOR GENERATION OF ADDITIONAL LINEAR CONSTRAINTS IN CRYPTANALYSIS OF SOME LIGHTWEIGHT CIPHERS.** In the paper, we propose a new technique that is aimed at algebraic cryptanalysis problems. Using this technique we construct additional linear equations over GF(2) which augment the system of algebraic equations presenting the cryptanalysis of the considered cipher. We use a SAT solver to generate such new linear equations. It was shown that the proposed technique allows one to increase the efficiency of guess-and-determine attacks which are based on the linearization sets. Effectiveness of the proposed technique was confirmed by computational experiments in which we considered the cryptanalysis of some variants of well-known stream cipher Trivium with a decreased number of steps in the initialization phase.
Keywords: *linearizing sets, guess-and-determine attack, quadratic systems over* GF(2)*, pseudo-Boolean optimization, Trivium.*

Belousova A. A., Tokareva N. N. **ON DIFFERENTIALS FOR THE MODIFICATION OF THE CIPHER SIMON BASED ON THE LAI — MESSI SCHEME.** We consider the block iterative cipher Simon based on the Feistel network and its modification based on the Lai — Messi scheme. Received estimates of differentials of the considered ciphers are compared. The results show that after 12 rounds, estimate of the maximum probability of a differential for the modified cipher Simon 32/64 without adding an orthomorphism is 2^{-24}, and with the addition of orthomorphism is between 2^{-24} and 2^{-63}, while the estimate of maximum probability for the original version is 2^{-36}.
Keywords: *Lay — Massey scheme, Feistel network, differential cryptanalysis.*

Bespalov M. S., Malkova K. M. **CODING INFORMATION BY WALSH MATRICES.** The representation of the general linear group $\mathrm{GL}(n, 2)$ by the automorphism subgroup $\mathrm{GL}(N, 2)$ under the multiplicative notation in its action in the space $\mathbb{R}^N$, where $N = 2^n$, is considered. Each matrix as an element of the group $\mathrm{GL}(n, 2)$ defines ordering: the group $\mathbb{Z}_2^n$ and its group of characters, which are popular in digital processing of information in the form of discrete Walsh functions. On the basis of the fast Walsh transform and this correspondence the authors created a software prototype of an automatic output signal coding system. The essence of the proposed software product is the number of possible permutations, which is calculated by the formula $(2^n - 2^0)(2^n - 2^1) \dots (2^n - 2^{n-1})$ for n-th order matrices. Based on the program, it is possible to organize a multi-channel system of reconfigurable decoders when transmitting hidden information over open communication channels.
Keywords: *discrete Walsh functions, code matrix, fast Walsh transform, Kronecker product.*

Gribanova I. A., Semenov A. A. **USING INVERSE BACKDOORS SETS TO CONSTRUCT GUESS-AND-DETERMINE ATTACKS ON HASH-FUNCTIONS MD4.** In the paper, we propose new preimage attacks on hash-functions MD4-k, $k > 39$. These attacks, related to the class of guess-and-determine attacks, are based on the idea of inverse backdoor set. We use SAT solvers to solve the cryptanalysis problems weakened by substitution of guessed bits to SAT encodings of the considered functions. The problem of search for an inverse backdoor set with relatively small complexity estimation is considered as a minimization problem of a special pseudo-Boolean function. To solve this problem, we apply several metaheuristic algorithms: tabu search algorithm, (1+1)-FEA_β, and a variant of genetic algorithm. These algorithms produce attacks on the considered functions with close complexity estimations. For the full-round compression function MD4 the best attack is constructed using the genetic algorithm.
Keywords: *preimage attack on hash function, guess-and-determine attacks, MD4, inverse backdoor sets, SAT.*

Doronin A. E., Kalgin K. V. **CONSTRUCTION OF CRYPTOGRAPHIC BOOLEAN FUNCTIONS USING SAT-SOLVERS.** In this paper, we propose a method for solving some cryptographic problems based on translation them into SAT-problems and application of SAT-solvers. We introduce construction of several formulas defining conditions of one-to-one property and differential uniformity of vectorial Boolean functions.
Keywords: *SAT-solvers, cryptography, Boolean functions.*

Kuznetsov A. A. **COMPUTATION OF REWRITING SYSTEMS IN FINITE GROUPS.** We present an algorithm computing the rewriting system R of a finite group generated by the fixed set of elements. We have proved that R is confluent and irreducible in this case. A necessary condition for the effective implementation of the algorithm is the availability of a fast procedure for multiplying elements in the group. For example, this group operation can be a composition of permutations, matrix multiplication, calculation of Hall's polynomials, etc. We study rewriting systems in finite two-generator groups of exponent five using the algorithm.
Keywords: *Burnside group, the rewriting system.*

Sofronova D.A., Kalgin K.V. **A COMPACT TRANSLATOR OF ALGORITHMS INTO BOOLEAN FORMULAS FOR USE IN CRYPTANALYSIS.** The program for converting the description of the cryptographic task to CNF is presented. A SAT solver establishes the truth of the formula and finds the values of the variables after that. Features of this development are universality, a small code size (300 lines of C++), easily modifiable and extensible implementation.
Keywords: *cryptanalysis, SAT-solver, guess-and-determine attack.*