УДК 519.719.2+512.542.74 DOI 10.17223/20710410/57/4

КРИПТОГРАФИЧЕСКИЕ СЛАБОСТИ АЛГОРИТМОВ ТИПА «ГИПЕРКУБ»

Д. И. Трифонов

Академия криптографии Российской Федерации, г. Москва, Россия

E-mail: d.arlekino@gmail.com

Рассмотрен класс блочных криптографических XSLP-алгоритмов, называемых «гиперкуб». Для алгоритмов данного класса получены оценки показателя рассеивания линейной среды для любого числа итераций. Показано, что при выборе преобразования P с использованием обобщённых графов де Брейна для рассматриваемых алгоритмов может не наступать лавинный эффект, вследствие чего ключ шифрования может быть определён с трудоёмкостью, существенной меньшей трудоёмкости тотального опробования ключей.

Ключевые слова: *XSLP-шифр, криптоанализ, линейный метод, показатель рассеивания, «гиперкуб».*

FLAWS OF HYPERCUBE-LIKE CIPHERS

D. I. Trifonov

Academy of Cryptograhy of Russian Federation, Moscow, Russia

A class of block XSLP cryptographic algorithms called “hypercube” is considered. These algorithms have a block size $n = n' \cdot m = n' \cdot m' \cdot k$ bits. A hypercube-like algorithm is an iterative block algorithm consisted of four main operations: (1) key addition (by XOR), (2) n'-bit S-box application, (3) block-diagonal diffusion matrix $\mathrm{diag}(A_1, \ldots, A_k)$, $A_i \in \mathrm{GF}(2)_{n'm',n'm'}$, multiplication with diffusion degree ρ, and (4) permutation. The main results are the following: 1) the idea of constructing linear correlations and probabilities of distribution of differences, determined by hypercube-like algorithms, has been described; 2) the linear environment propagation index for any number of rounds has been evaluated; 3) the relevance of branch number $\theta(r)$ for differential trails probability and correlation of linear trails for any $r \in \mathbb{N}$, $r \geqslant 2$, rounds has been formally represented; 4) for hypercube-like algorithms, it is shown that when constructing a P-transform using de Bruijn graphs, the avalanche effect may not occur, which means that the (time) complexity of determining the encryption key will be much less than the exhaustive key search (time) complexity. Let $n = n'(m')^d$ and $\mathrm{P} : V_n \to V_n$ affect $a = (a_0, \ldots, a_{m-1}) \in V_n$, $a_i \in V_{n'}$, as follows. Numbers $l \in \{0, \ldots, (m')^d - 1\}$ of $a_l \in V_{n'}$ in $a \in V_n$ are considered as $l = j_0 + j_1 m' + \ldots + j_{d-1}(m')^{d-1}$, $j_t = 0, \ldots, m' - 1$, $t = 0, \ldots, d - 1$. Let the mapping P is defined as $\mathrm{P}(a) = \mathrm{P}(a_0, \ldots, a_{(m')^d-1}) = (a_{\tau(0)}, \ldots, a_{\tau((m')^d-1)})$, $\tau \in S_{(m')^d}$, $\tau(l) = \tau(j_0, \ldots, j_{d-1})$, $l = 1, \ldots, (m')^d$. In the case $d = 3$ it is obtained that if P is rotation of hypercube, i.e., $\tau(j_0, j_1, j_2) = (j_1, j_2, j_0)$, then $\theta(r) \leqslant t(r)$, $t(1) = m'$, $t(r) = ((m')^2 + m')\,[r/2] + m'(r \bmod 2)$, $r \geqslant 2$. In the case $\tau(i_0, i_1, i_2) = (i_0, i_1 + i_0 \bmod m', i_2 + i_0 \bmod m')$ we obtain $\theta(r) = \theta(r - 4) + \rho^2$, $\theta(1) = 1$, $\theta(2) = \rho$, $\theta(3) = 2\rho - 1$, $r \in \mathbb{N}$, $r > 4$.

Keywords: *XSLP-ciphers, cryptoanalysis, linear method, branch numbers, hypercube structure.*

Введение

В [1] отмечено, что благодаря появлению линейного метода криптографического анализа в отечественной и зарубежной практике сформировался уже устоявшийся подход к построению блочных шифров: в качестве нелинейных необходимо использовать преобразования, обеспечивающие необходимые характеристики относительно линейного и разностного методов, а в качестве линейных преобразований — преобразования с достаточно высокими рассеивающими свойствами. По такому принципу построено много криптографических алгоритмов, среди которых можно отметить российский стандарт блочного шифрования «Кузнечик» [2].

Вместе с тем выбор криптографически качественных примитивов далеко не всегда означает построение блочного алгоритма с высокими криптографическими свойствами (см., например, [1, 3]). В данной работе показано, что для XSLP-алгоритмов с использованием в качестве преобразования L блочно-диагональной матрицы с максимально рассеивающими матрицами на диагонали, а в качестве преобразования P коммутации, построенной с использованием обобщённых графов де Брейна, существуют эффективные схемы согласования, позволяющие строить эффективные методы определения ключа.

1. Определения и обозначения

В работе используются следующие определения и обозначения [4, 5]:

— $\mathbb{Z}_n$ — кольцо вычетов по модулю n;
— $\mathbb{F}_{m,n}$ — множество всех матриц из m строк и n столбцов с элементами из поля $\mathbb{F}$; в случае поля $\mathbb{F}_2$ применяется обозначение $(\mathbb{F}_2)_{n,n} = M_n$;
— $S(A)$ — симметрическая группа подстановок на непустом множестве A; в случае $A = \{0, 1, \ldots, m-1\}$ применяется обозначение S_m.

Определение 1. Под наступлением лавинного эффекта на t-й итерации блочного криптографического алгоритма будем понимать существенную зависимость всех бит обрабатываемого информационного блока на t-й итерации от всех входных бит, $t \in \mathbb{N}$.

Определение 2 [6]. Пусть $m > 1$, $r > 1$ — натуральные числа, $n = m^r$. Множество вершин графа де Брейна порядка r на $n = m^r$ вершинах составляют все r-мерные векторы вида $(\nu_0, \nu_1, \ldots, \nu_{r-1})$, $\nu_i \in \mathbb{Z}_m$, $i = 0, \ldots, r-1$. Из вершины $(\nu_0, \nu_1, \ldots, \nu_{r-1})$ исходит m дуг в вершины $(\varepsilon, \nu_0, \nu_1, \ldots, \nu_{r-2})$; в неё также входит m дуг из вершин $(\nu_1, \ldots, \nu_{r-1}, \varepsilon)$, где ε — любой элемент $\mathbb{Z}_m$.

Замечание 1. В условиях определения 2 в графе де Брейна петли имеются в точности на m вершинах $(\nu, \ldots, \nu) \in \mathbb{Z}_m$.

Определение 3 [6]. Пусть n, r — натуральные числа, $n > 1$, $r \geqslant 1$. Ориентированный граф на n вершинах называется ∂-графом (или обобщённым графом де Брейна) порядка r, если для любой пары его вершин существует единственный направленный путь длины r из одной вершины в другую.

Замечание 2. Если в ∂-графе Γ порядка r изменить направление по каждой его дуге на противоположное, то получающийся граф $\overline{\Gamma}$ также будет ∂-графом порядка r.

2. Постановка задачи

В [7, п. 9.7.1] для построения блочных алгоритмов с высокими криптографическими характеристиками предлагается использовать структуру «гиперкуб» (d-мерный куб). Каждый алгоритм, имеющий структуру «гиперкуб», по сути представляет собой XSLP-алгоритм со следующими параметрами: n — размер блока в битах; d — размерность гиперкуба; m — число нелинейных преобразований (подстановок) на n'-подвекторах; m' — размер стороны гиперкуба; $n = n' \cdot m = n' \cdot m' \cdot k$; r — число итераций. Будем полагать $k = (m')^{d-1}$, то есть $n = n'(m')^d$, $m = m' \cdot k = (m')^d$.

Обрабатываемый данным алгоритмом информационный блок представляется следующим образом. Компоненты вектора $(a_1, \ldots, a_m) \in V_n$, $a_i \in V_{n'}$, $i = 1, \ldots, m$, записываются в ячейки d-мерного куба последовательно по всем граням. Так, в случае $d = 2$ запись происходит в таблицу размера $m' \times m'$.

При реализации алгоритмов из предлагаемого семейства на каждой итерации используются следующие преобразования:

Наложение ключа $\mathrm{X}[K] : V_n \to V_n$, где $\mathrm{X}[K](a) = K \oplus a$; $a, K \in V_n$.

Нелинейное преобразование $\mathrm{S} : V_n \to V_n$, $\mathrm{S}(a) = \mathrm{S}(a_0, \ldots, a_{m-1}) = (\pi_0(a_0), \ldots, \pi_{m-1}(a_{m-1}))$, $a_i \in V_{n'}$, $\pi_i \in S(V_{n'})$, $i = 0, \ldots, m-1$.

Рассеивающее преобразование $\mathrm{L} : V_n \to V_n$,

$$\mathrm{L}(a) = (a_0, \ldots, a_{m'-1}, a_{m'}, \ldots, a_{2m'-1}, \ldots a_{m'k-1}) = (b_0, \ldots, b_{m'-1}, b_{m'}, \ldots, b_{2m'-1}, \ldots b_{m'k-1}),$$

$a_i \in V_{n'}$, $i = 1, \ldots, m' \cdot k$ при этом $(b_{m'(i-1)}, \ldots, b_{im'-1}) = (a_{m'(i-1)}, \ldots, a_{im'-1})A_i$, где $A_i \in \mathrm{GF}(2)_{n'm',n'm'}$, $i = 1, \ldots, k$; $a_j, b_j \in V_{n'}$, $j = 0, \ldots, m'k-1$; $a_i \in V_{n' \cdot m'}$, $(a_1 \| \ldots \| a_k) \in V_n$;

Коммутация $\mathrm{P} : V_n \to V_n$, действует на вектор $a = (a_0, \ldots, a_{m-1}) \in V_n$, $a_i \in V_{n'}$, следующим образом. При $m = (m')^d$ представим номер $l \in \{0, \ldots, (m')^d - 1\}$ подвектора $a_l \in V_{n'}$ вектора $a \in V_n$ в виде m'-ичной записи $l = j_0 + j_1 m' + \ldots + j_{d-1}(m')^{d-1}$, $j_t = 0, \ldots, m'-1$, $t = 0, \ldots, d-1$, которой будем ставить во взаимно-однозначное соответствие набор $(j_0, j_1, \ldots, j_{d-1})$. Тогда $\mathrm{P}(a) = \mathrm{P}(a_0, \ldots, a_{(m')^d-1}) = (a_{\tau(0)}, \ldots, a_{\tau((m')^d-1)})$, где $\tau \in S_{(m')^d}$. $\tau(l) = \tau(j_0, \ldots, j_{d-1}) = (j_1, j_2, \ldots, j_0)$, $l = 1, \ldots, (m')^d$.

В рассматриваемой конструкции XSLP-алгоритма ключевую роль играет выбор перестановки на номерах подвекторов (коммутации) τ. В [7, п. 9.7.1] предлагается использовать коммутацию вида

$$(j_0, \ldots, j_{d-1}) \xrightarrow{\tau} (j_1, j_2, \ldots, j_0). \tag{1}$$

Такой выбор явился следствием глубокой проработанности вопроса использования данной коммутации в SP-сетях [8]. В [8] также указывается на то, что в SP-сетях степень рассеивания для преобразования P определяется по лавинному эффекту. Отметим, что для SP-сети преобразование P задаётся коммутацией на номерах координат вектора. Согласно [8, теорема 3], оптимальными в этом отношении являются подстановки $\mathrm{P} \in S_n$, для которых переходы $(i, \mathrm{P}(i))$, $i = 1, \ldots, n$, содержатся среди дуг обобщённых графов де Брейна [6].

Во введённых определениях, исходя из [8, теорема 3], с точки зрения качественного представления о рассеивании наиболее предпочтительными при $n = m^r$ являются коммутации $\mathrm{P} \in S_n$, для которых $\Gamma(P)$ или $\Gamma(P^{-1})$ является ∂-графом порядка r, в частности графом де Брейна, $r \in \mathbb{N}$. Данные рассуждения также переносятся на рассматриваемый в настоящей работе случай $n = n'(m')^d$, когда преобразование P используется для рассеивания подвекторов длины n'. В этом случае дугами графа де

Брейна Γ_0 на множестве $\{0, 1, \ldots, (m')^d - 1\}$ являются пары вершин

$$\big((i_0, i_1, \ldots, i_{d-1}), (i_1, \ldots, i_{d-1}, j)\big) : i_0, i_1, \ldots, i_{d-1}, j \in \{0, 1, \ldots, m' - 1\}.$$

В «канонической» SP-сети [8] в качестве коммутации $P_0 \in S_{m^r}$, для которой $\Gamma(P_0^{-1}) = \overline{\Gamma}_0$, рекомендуется брать подстановку P_0, задаваемую коммутацией вида (1):

$$P_0(i) = P_0(i_0, \ldots, i_{r-1}) = (i_1, \ldots, i_{r-1}, i_0).$$

При $d = 2$ коммутация τ вида (1) хорошо зарекомендовала себя при использовании в блочных криптографических алгоритмах и хэш-функциях (выделим в первую очередь алгоритмы AES [9] и Стрибог [10]). Вместе с тем далее показано, что при $d = 3$ коммутация τ вида (1) обладает недостаточными рассеивающими свойствами. Данное обстоятельство приводит к отсутствию лавинного эффекта и как следствие — к возможности построения эффективных схем согласования для определения ключа алгоритма.

Замечание 3. Коммутация P вида (1) для XSLP-алгоритма со структурой «гиперкуб» соответствует с точностью до переобозначений повороту d-мерного куба относительно одной из осей. Согласно [7, с. 145], при $d > 1$ данное преобразование является «оптимальным» в смысле обеспечения существенной зависимости бит промежуточного блока от входных данных.

Пример 1. Пример преобразования P в случаях $d = 2$ и 3 представлен на рис. 1.

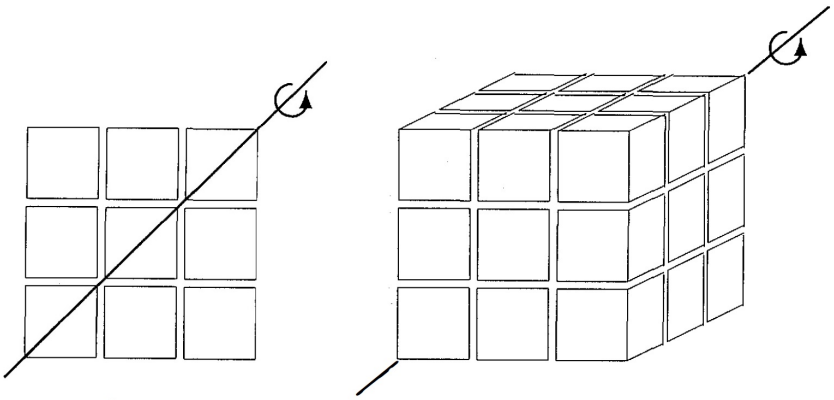

Рис. 1. Действие коммутации P в случаях $d = 2$ и 3

В случае $d = 3$ нижняя строка подстановки τ имеет следующий вид:

$$1, 10, 19, 2, 11, 20, 3, 12, 21, 4, 13, 22, 5, 14, 23, 6, 15, 24, 7, 16, 25, 8, 17, 26, 9, 18, 27$$

Графически перестановка вида (1) номеров подвекторов, записанных в 3-мерный куб, приведена на рис. 2. Для большей наглядности вместо подстановки τ приведена подстановка τ^{-1}, нижняя строка которой имеет следующий вид:

$$1, 4, 7, 10, 13, 16, 19, 22, 25, 2, 5, 8, 11, 14, 17, 20, 23, 26, 3, 6, 9, 12, 15, 18, 21, 24, 27.$$

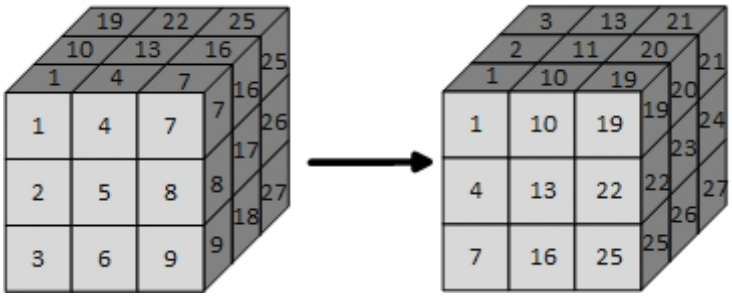

Рис. 2. Коммутация τ^{-1} при $d = 3$

3. Примеры коммутаций

Приведём примеры коммутаций τ вида (1), используемых в различных криптографических алгоритмах. На некоторые из рассмотренных примеров автору указали А. В. Анашкин и Ф. М. Малышев.

Алгоритм LOKI91 [11, 12]. Данный алгоритм схож по своему строению с алгоритмом DES [13]. В алгоритме LOKI91 используется перестановка $t \in S_{32}$, для которой нижняя строка подстановки t^{-1} имеет следующий вид:

$$31, 23, 15, 7;\ 30, 22, 14, 6;\ 29, 21, 13, 5;\ 28, 20, 12, 4;$$
$$27, 19, 11, 3;\ 26, 18, 10, 2;\ 25, 17, 9, 1;\ 24, 16, 8, 0.$$

Естественней здесь выглядит подстановка $\sigma \in S_{32}$, $\sigma(i) = t(32 - i)$, $i = 0, 1, \ldots, 31$. Для задания подстановки σ верхняя строка разбивается на четыре восьмёрки подряд идущих чисел, а нижняя — на восемь четвёрок подряд идущих чисел. Тогда верхняя строка параметризуется парами (i, j), а нижняя — парами (j, i), $i = 0, \ldots, 3$, $j = 0, \ldots, 7$. В этом случае подстановка σ определяется правилом $\sigma(i, j) = (j, i)$.

Алгоритм SAFER K-64 [14, 15]. Данный алгоритм представляет собой SP-сеть. В алгоритме используется подстановка $t \in S_8$, нижняя строка которой имеет вид $(0, 4, 1, 5, 2, 6, 3, 7)$. Подстановка t может быть задана как подстановка σ в алгоритме LOKI91, а именно: $t(i, j) = [j, i]$, где $(i, j) = 2i + j$, $[j, i] = 4j + i$, $i = 0, \ldots, 3$, $j = 0, 1$. Если $i \in \{0, \ldots, 7\}$ представлять в двоичном виде $i = (i_0, i_1, i_2) = i_0 \cdot 2^2 + i_1 \cdot 2 + i_2$, то $i = (i_0, i_1)$, $j = i_2$ и $t(i_0, i_1, i_2) = (i_2, i_0, i_1)$.

Нетрудно убедиться, что ориентированный граф Γ на множестве вершин $\{0, \ldots, 7\}$ с дугами $i \to t^{-1}(i)$, $i \to t^{-1}(i) \pm 1$, где знак «+» берётся при чётном $t^{-1}(i)$ и знак «−» иначе, является графом де Брейна.

Далее приведём два примера криптографических алгоритмов с длиной блока $n = n'(m')^2$, в которых коммутация τ рассматривается в несколько обобщённом виде, а именно:

$$(j_0, \ldots, j_{d-1}) \xrightarrow{\tau} (j_1, j_2, \ldots, T(j_0, j_1, \ldots, j_{d-1})), \tag{2}$$

где $T : \mathbb{Z}_{m'}^d \to \mathbb{Z}_{m'}$ — некоторая функция.

Алгоритм AES [9]. В криптографическом алгоритме AES с длиной блока $128 = 8 \cdot 4^2$ бит и длиной ключа 256 бит в качестве рассеивающего преобразования P (ShiftRow Transformation) используется следующая подстановка:

$$\mathrm{P} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 13 & 10 & 7 & 4 & 1 & 14 & 11 & 8 & 5 & 2 & 15 & 12 & 9 & 6 & 3 \end{pmatrix}.$$

Тогда, представив числа из множества $\{0,\dots,15\}$ в виде $i = i_0 + i_1 \cdot 4$, $i_0, i_1 = 0,1,2,3$, получаем, что преобразование P имеет вид

$$(i_0, i_1) \to (i_0,\ (i_1 - i_0) \bmod 4).$$

Строго говоря, данное преобразование не является преобразованием вида (2). Для того чтобы привести его к виду (2), произведём перенумерацию подвекторов длины n', которые являются входами для соответствующих нелинейных преобразований. Перенумерация заключается в переходе к новым младшим цифрам в номере с сохранением старших. Другими словами, вместо (i_0, i_1) рассматриваются (j_0, j_1), $j_1 = i_1$, $j_0 = i_1 + i_0$. Тогда номера подвекторов длины n' изменятся по следующей подстановке:

$$\tau = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 1 & 2 & 3 & 5 & 6 & 7 & 4 & 10 & 11 & 8 & 9 & 15 & 12 & 13 & 14 \end{pmatrix},$$

а коммутация AES примет вид

$$\mathrm{P}(i_0, i_1) = \tau^{-1}(i_1, T_{i_1}(i_0)),$$

где

$$T_0 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 3 & 2 & 1 \end{pmatrix},\ T_1 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \end{pmatrix},\ T_2 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 3 \end{pmatrix},\ T_3 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 \end{pmatrix}.$$

Алгоритм LUCIFER [16]. Здесь рассматривается изменённая версия алгоритма Люцифер (иногда обозначается Lucifer (v1)), которая была выдвинута компанией IBM в качестве национального стандарта США. Алгоритм представляет собой 8-раундовую SP-сеть и имеет длину блока 128 бит и длину ключа 256 бит. При длине блока $n = (m')^2$ в качестве подстановки $t \in S_{(m')^2}$ используется подстановка $t(i_0, i_1) = (i_1, w(i_1) + i_0)$, $i_0, i_1 \in \{0,\dots,m'-1\}$, сложение производится по модулю m' и $w \in S_{m'}$.

Перенумерацией, рассмотренной для алгоритма AES, можно прийти к аналогичному виду коммутации:

$$(j_0, j_1) \to (j_1, 2j_1 - j_0).$$

4. Показатель рассеивания

Согласно [17, 8, 18], линейный метод нацелен на получение вероятностного линейного соотношения шифрпреобразования

$$F : V_N \times V_K \to V_M,\quad V_N \times V_K \ni (a, z) \mapsto b = F(a, z) \in V_M \tag{3}$$

общего вида, задаваемого функциональной схемой, которая представляется последовательностью выполняемых в определённом порядке линейных и нелинейных преобразований. Для блочного шифрования в (3) $M = N$, $a \in V_N$ и $b \in V_N$ — соответствующие блоки открытого и шифрованного текстов, $z \in V_K$ — ключ. Предполагается, что биты ключа складываются по модулю 2 с соответствующими промежуточными знаками шифрпреобразования.

Последовательность преобразований функциональной схемы можно представлять в виде условных или безусловных шагов алгоритма, не содержащего операторов перехода на предыдущие операторы. Пусть нелинейные операции в этой программе

$$f_i : V_{n_i} \to V_{m_i},\ x_i \mapsto y_i = f_i(x_i),\ i = 1,\dots,k, \tag{4}$$

следуют в указанном порядке. Значения переменных $x_i \in V_{n_i}$, $y_i \in V_{m_i}$ и $b \in V_M$ однозначно задаются значениями $a \in V_N$ и $z \in V_K$. В промежутках между операциями (4) (как и в начале и в конце программы) могут быть линейные преобразования.

Основным этапом в линейном методе применительно к шифрпреобразованию (3) является нахождение трёх конкретных вектор-столбцов L', L и $L'' \neq 0$ размеров N, K и M соответственно, для которых желательно, чтобы булева случайная величина $\eta = aL' + zL + bL''$ имела как можно большее значение $\delta = |2\,\mathsf{P}[\eta = 0] - 1|$ для как можно большего числа ключей $z \in V_K$, где вероятность $\mathsf{P}[\cdot]$ вычисляется при случайном равновероятном выборе вектора $a \in V_N$, ключ $z \in V_K$ фиксирован.

Случайная величина η определяется по набору вектор-столбцов

$$\pounds = ((l'_i, l''_i), i = 1, \ldots, k)$$

размеров соответственно n_i и m_i как

$$\eta = \sum_{i=1}^{s} (x_i l'_i + y_i l''_i). \tag{5}$$

Чтобы такая сумма приняла вид $aL' + zL + bL''$ для каких-либо вектор-столбцов L', L и $L'' \neq 0$, набор $\pounds$ должен удовлетворять специальным условиям согласованности. Для их формулировки компоненты l'_i припишем на функциональной схеме в виде входных пометок к соответствующим компонентам x_i, а компоненты l''_i аналогично припишем к компонентам y_i в виде выходных пометок. Расстановка пометок в виде элементов из GF(2) распространяется и на каждую линейную операцию в шифрпреобразовании (3). При этом если $g : V_n \to V_m$, $V_n \ni x \mapsto y = xg \in V_m$, — одна из линейных операций в шифрпреобразовании (3) и входные пометки g образуют вектор-столбец α' размера n, а выходные пометки g — вектор-столбец α'' размера m, то в целях обеспечения равенства $x\alpha' = y\alpha''$ условие согласованности (в соответствии с $x\alpha' = xg\alpha''$) предполагает выполнение равенства $\alpha' = g\alpha''$. В частности, пометки вокруг сумматора

$$(x_1, \ldots, x_n) \mapsto x_1 + \ldots + x_n = y_1$$

одинаковы, а пометки вокруг узла размножения

$$x \mapsto (x_1, \ldots, x_n) = (y_1, \ldots, y_m)$$

в сумме дают нуль. При $n = m$ и $g = \mathrm{id}_{V_n}$ обязательно $\alpha' = \alpha''$. Последнее означает, что если выход некоторого отображения в шифрпреобразовании (3) является входом какого-то одного отображения, то к этим выходу и входу приписываются одинаковые пометки.

Пара (l'_i, l''_i) в наборе $\pounds$, такая, что $l'_i \neq 0$, $l''_i = 0$ или $l'_i = 0$, $l''_i \neq 0$, а f_i биективно, считается дефектной. Тогда слагаемые $x_i l'_i + y_i l''_i$ в (5) при случайном равновероятном $x_i \in V_{n_i}$ обладают равномерным распределением на GF(2), то есть имеет максимальную неопределённость.

Пусть W — совокупность всех согласованных наборов пар вектор-столбцов $\pounds = ((l'_i, l''_i), i = 1, \ldots, k)$ без дефектов, в которых хотя бы один вектор-столбец ненулевой (данный набор будем также называть $\pounds$-путём). Обозначим

$$\theta_{\pounds} = |\{i = 1, \ldots, k \ : l''_i \neq 0\}|\,.$$

Для уменьшения неопределённости суммы (5) приходится следить за уменьшением неопределённости отдельных слагаемых. Она минимальна при $l'_i = 0$, $l''_i = 0$. В этой связи становится важной характеристика

$$\theta = \min_{\pounds \in W} \theta_{\pounds} \tag{6}$$

для шифрпреобразования (3), точнее, для линейной среды этого шифрпреобразования, образованной всеми линейными преобразованиями соответствующей функциональной схемы. Замена совокупности нелинейных отображений (4) на любые другие отображения при тех же n_i, m_i никак не скажется на величине θ. Чем больше θ, тем соответствующий шифр является более стойким к линейному методу и тем легче можно этого добиваться путём специального выбора нелинейных отображений (4). Более тщательный выбор последних отображений позволяет также (в большинстве случаев) повышать стойкость к другим методам криптографического анализа.

5. Основной результат

Не ограничивая общность, для простоты изложения будем рассматривать случай $d = 3$, $n = n'(m')^3$. Приведённые далее рассуждения могут быть без труда обобщены на случай $d > 3$.

Пусть также коммутация τ на номерах координат имеет вид

$$(j_0, j_1, j_2) \to (j_1, j_2, j_0), \quad j_0, j_1, j_2 \in \{0, 1, \ldots, m' - 1\}, \tag{7}$$

где при фиксированных j_1, j_2 число j_0 принимает всевозможные значения из множества $\{0, 1, \ldots, m' - 1\}$. Как и в примере 1, в рассматриваемом алгоритме преобразование P будем отождествлять с поворотом влево 3-мерного куба относительно оси z в системе координат (x, y, z).

Покажем, что в данном случае для XSPL-алгоритма при любом числе итераций не наступает существенная зависимость бит промежуточного блока от каждого бита входного блока. Эта слабость приводит к появлению эффективной схемы согласования, что существенно влияет на стойкость рассматриваемого криптографического алгоритма.

Следуя работе [19], введём специальное представление XSLP-алгоритма в виде графа. Данное представление тесно связано с оценкой эффективности линейного метода криптографического анализа. В частности, теоретико-графовый подход хорошо зарекомендовал себя при вычислении показателя рассеивания линейной среды шифрпреобразования (более подробно см. [19, 20]).

Пусть $\pounds = ((l'_{ij}, l''_{ij}), i = 1, \ldots, r,\ j = 0, \ldots, m - 1)$ — произвольный ненулевой согласованный набор столбцов без дефектов, входящий в определение (6) показателя θ. Столбец l'_{ij} состоит из n' бит меток входов подстановки π_{ij} (без ограничения общности будем считать, что $\pi_{ij} = \pi$), а столбец l''_{ij} — из n' меток выходов этой же подстановки. Отсутствие дефектов означает, что $l'_{ij} = 0$ тогда и только тогда, когда $l''_{ij} = 0$, $i = 1, \ldots, r$, $j = 0, \ldots, m - 1$. Согласованность $\pounds$ означает, что

$$l''_i = \widehat{\mathrm{P}}(\mathrm{L}(l'_{i+1})),\ i = 0, 1, \ldots, r - 1,$$

где

— столбец l''_i составлен из столбцов $l''_{i,j}$, $j = 0, \ldots, m - 1$;
— столбец l'_{i+1} составлен из столбцов $l'_{i+1,j}$, $j = 0, \ldots, m - 1$;

— $L \in \mathrm{GF}(2)_{n'm,n'm}$ — блочно-диагональная матрица, составленная из матриц B_{i,j_1}, $j_1 = 1,\ldots,k$, $i = 1,\ldots,r-1$, которые имеют показатель рассеивания ρ [19] относительно разбиения вектора длины $n' \cdot m'$ на подвектора длины n';

— $\widehat{\mathrm{P}} \in \mathrm{GF}(2)_{n'\cdot m,n'\cdot m}$ — подстановочная матрица, осуществляющая перестановку n'-векторов в соответствии с коммутацией τ. При этом меняются местами только n'-векторы, соответствующие подстановкам

$$\pi_{i,(j_0,j_1,j_2)},\ \pi_{i,(j_1,j_2,j_0)},$$

для всех $i = 1,\ldots,r$ и всех $j = j_0 + j_1 m' + j_2 (m')^2$, $j_0, j_1, j_2 \in \{0,1,\ldots,m'-1\}$.

Набору $\pounds$ поставим в соответствие неориентированный граф Γ. *Вершинами* графа Γ, располагающимися в следующих сверху вниз $r-1$ ярусах, являются те блоки $B_{i,(j_1,j_2)}$, $i = 1,\ldots,r-1$, $j_1, j_2 = 0,\ldots,m'-1$, для которых $l''_{i,*(j_1,j_2)} \neq 0$, а значит, и $l'_{i,*(j_1,j_2)} \neq 0$. Здесь столбец $l''_{i,*(j_1,j_2)}$ ($l'_{i,*(j_1,j_2)} \neq 0$) составлен из столбцов $l''_{i,(j_0,j_1,j_2)}$ (соответственно $l'_{i,(j_0,j_1,j_2)}$), $j_0 = 0,\ldots,m'-1$.

Ребром графа Γ будем называть пару вершин $(B_{i,(j_1,j_2)}, B_{i+1,(j_2,j_0)})$ при некотором $j_0 = 0,\ldots,m'-1$, таком, что $l'_{i+1,*(j_2,j_0)} \neq 0$.

Дополнительно к каждой вершине $B_{1,(j_1,j_2)}$ верхнего яруса добавим

$$[[l''_{1,*(j_1,j_2)}]] = \left|\{j_0 = 0,\ldots,m'-1 : l''_{1,(j_0,j_1,j_2)} \neq 0\}\right|$$

рёбер, а к каждой вершине $B_{r-1,(j_1,j_2)}$ нижнего яруса — $[[B^{-1}_{r-1,(j_1,j_2)} l''_{1,*(j_1,j_2)}]]$ рёбер. После этого величина $\theta_{\pounds}$ будет совпадать с числом рёбер в графе Γ.

Для удобства графического представления графа Γ будем изображать его следующим образом. Все $(m')^2$ вершин графа Γ запишем в таблицу из m' строк и столбцов. Запись в таблицу будем производить по строкам слева направо. Таким образом, граф Γ имеет вид, приведённый на рис. 3.

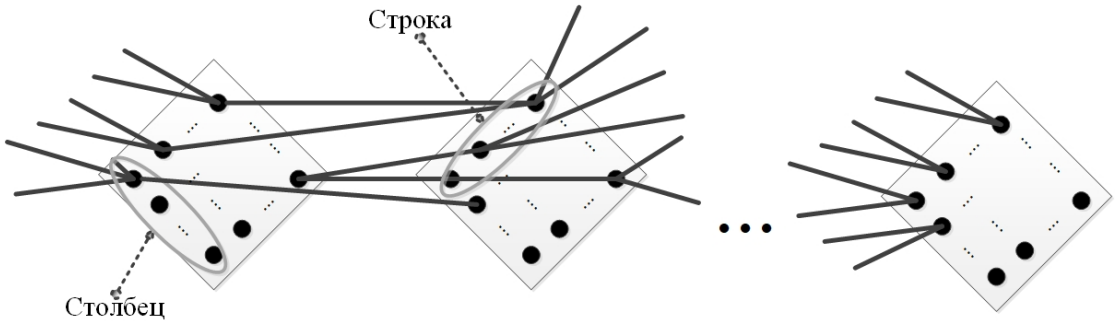

Рис. 3. Общий вид графа Γ для XSLP-алгоритма с длиной блока $n = n'(m')^3$

Граф Γ для XSLP-алгоритма, состоящего из $r \in \mathbb{N}$ итераций, будем обозначать $\Gamma^{(r)}$. Часть графа $\Gamma^{(r)}$, соответствующую i-й итерации, будем называть $(i-1)$-м слоем, $i \leqslant r$. Слои в графе $\Gamma^{(r)}$ занумеруем справа налево начиная с нуля. Количество неизолированных вершин в i-м слое обозначим через M_i, $0 \leqslant i \leqslant r-2$.

5.1. Показатель рассеивания алгоритма типа гиперкуб

Лемма 1. Для любого $r \in \mathbb{N}$, $r \geqslant 2$, и графа $\Gamma^{(r)}$ каждая неизолированная вершина графа i-го слоя смежна лишь с такими вершинами $(i+1)$-го слоя, которые стоят в одном столбце и в разных строках $(i+1)$-го слоя, $0 \leqslant i \leqslant r-2$.

Доказательство. Рассмотрим произвольную неизолированную вершину графа $\Gamma^{(i)}$. Из неё выходят рёбра с номерами $(j_0, j_1, j_2) = j_0 + j_1 m' + j_2 (m')^2$; $j_1, j_2 \in \in \{0, \ldots, m'-1\}$ фиксированы, j_0 принимает все значения из множества $\{0, \ldots, m'-1\}$. Согласно определению коммутации τ (7), данные рёбра (подвекторы длины n') будут переставляться в соответствии с правилом

$$\tau(j_0, j_1, j_2) = (j_1, j_2, j_0).$$

Для завершения доказательства необходимо заметить, что ребро с номером (j_0, j_1, j_2) инцидентно вершине, стоящей с строке с номером j_2 и столбце с номером j_1. ■

Следствие 1. В условиях леммы 1, вершины i-го слоя графа $\Gamma^{(r)}$, лежащие в j-й строке, смежны лишь с такими вершинами $(i+1)$-го слоя, которые лежат в j-м столбце, $j = 1, 2, \ldots, m'$.

Лемма 2. Для любого $r \in \mathbb{N}$, $r \geqslant 2$, и графа $\Gamma^{(r)}$ выполнены следующие неравенства:

$$M_i + M_{i+2} \geqslant \rho, \quad 0 \leqslant i \leqslant r-3. \tag{8}$$

Доказательство. Рассмотрим три подряд идущих слоя графа $\Gamma^{(r)}$. Нетрудно убедиться в том, что неравенство (8) обращается в равенство при $M_{i+1} = 1$. Так как в графе $\Gamma^{(r)}$ отсутствуют петли и кратные дуги, а также в силу $\rho(A_i) = \rho$, $i = 1, \ldots, k$, получаем, что единственная вершина $(i+1)$-го слоя инцидентна по крайней мере ρ рёбрам, а значит, смежна по крайней мере с ρ вершинами, находящимися в i-м и $(i+2)$-м слоях. Следовательно, $M_i + M_{i+2} \geqslant \rho$. Графическая интерпретация ситуации, когда неравенство (8) обращается в равенство, при $m' = 3$ представлена на рис. 4.

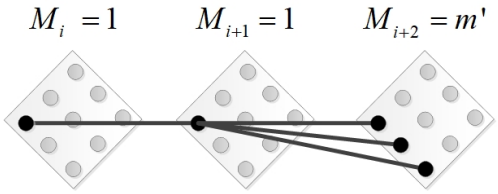

Рис. 4. Пример графа $\Gamma^{(4)}$, для которого $M_i + M_{i+2} = \rho$ при $m' = 3$ и $\rho = m' + 1$

Если предположить, что $M_i + M_{i+2} < \rho$, то неизолированная вершина $(i+1)$-го слоя будет инцидентна менее чем ρ рёбрам, что противоречит условию $\rho(A_i) = \rho$, $i = 1, \ldots, k$. ■

Теорема 1. Пусть $\rho = m' + 1$. Тогда для показателя рассеивания $\theta = \theta(r)$ линейной среды XSLP-алгоритма на $r \in \mathbb{N}$, $r \geqslant 2$, итераций справедлива следующая оценка:

$$\theta(r) \leqslant t(r), \quad t(1) = m',$$
$$t(r) = ((m')^2 + m') \left[\frac{r}{2}\right] + m'(r \bmod 2), \quad r \geqslant 2.$$

Доказательство. Используем лемму 1, а также тот факт, что для любого графа $\Gamma^{(r)}$ выполнено $\theta(r) \leqslant \upsilon(\Gamma^{(r)})$, где $\upsilon(\Gamma^{(r)})$ — количество рёбер в графе $\Gamma^{(r)}$. Построим граф $\Gamma^{(r)}$, такой, что $t(r) = \upsilon(\Gamma^{(r)})$.

Выберем $l \in \{1, \ldots, m'\}$. Начнём построение графа с первого слоя, в котором m' неизолированных вершин находятся в l-м столбце. Из каждой неизолированной

вершины выходит по одному ребру, причём все рёбра смежны лишь с вершинами в l-й строке. Второй слой состоит из m' неизолированных вершин, находящихся в l-й строке. В каждую вершину входит одна и выходит m' рёбер таким образом, что третий слой содержит m' неизолированных вершин, лежащих в l-м столбце. Таким образом, в каждую неизолированную вершину третьего слоя входит m' рёбер и выходит одно ребро. Далее граф строится по индукции аналогичным образом.

Пример описываемого графа $\Gamma^{(r)}$ при $m' = 3$ приведён на рис. 5. ■

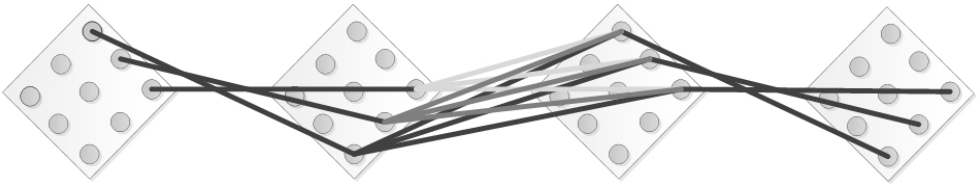

Рис. 5. Граф XSLP-алгоритма с длиной блока $n = n'(m')^3$ при $m' = 3$, для которого $\upsilon(\Gamma^{(r)}) = t(r)$, $r = 1, 2, \ldots$

5.2. Один подход к определению ключа алгоритмов типа «гиперкуб»

Более подробно остановимся на существенной слабости рассматриваемого XSLP-алгоритма, имеющего структуру «гиперкуб». Для этого приведём подход к методу определения ключа для данного алгоритма и оценим его трудоёмкость. Пусть для XSLP-алгоритма на каждой итерации используются случайно равновероятно выбранные ключи $K \in V_n$. Разобъём XSLP-алгоритм на m' параллельно действующих алгоритмов меньшей размерности. При этом каждый из таких алгоритмов получается, согласно доказательству теоремы 1, путём выбора номера $l \in \{1, \ldots, m'\}$ столбца, содержащего неизолированные вершины в первом слое соответствующего графа. Тогда, используя результаты и терминологию теоремы 1, получаем, что в каждом алгоритме меньшей размерности при любом числе итераций $r \in \mathbb{N}$ вместо $n \cdot r$ бит ключа будет использоваться только $n' \cdot t(r)$ бит.

Учитывая, что параллельно действующие алгоритмы меньшей размерности между собой никак не связаны, получаем следующую оценку трудоёмкости метода определения ключа:

$$m'(n' \cdot t(r)),$$

что существенно меньше трудоёмкости полного опробования $n \cdot r$.

5.3. Ещё один вариант коммутации для алгоритмов типа «гиперкуб»

Покажем, что, вообще говоря, выбор преобразования P в XSLP-алгоритмах представляет собой нетривиальную задачу, а обоснование криптографических качеств таких алгоритмов в целом — трудоёмкий процесс, в котором можно легко пойти по ложному пути.

Рассмотрим вопрос выбора коммутации в описанном XSLP-алгоритме типа «гиперкуб» с длиной блока $n = n'(m')^3$. За основу выбран принцип, используемый в алгоритме типа «гиперкуб» при $d = 2$ и $n = n'(m')^2$. Рассмотрев граф Γ такого алгоритма, можно убедиться, что он состоит из m' вершин, в каждую из которой в общем случае приходит не более m' рёбер (каждое ребро выходит из своей вершины) и выходит не более m' вершин (опять-таки в разные вершины).

Сохранение этого принципа при $d = 3$ в явном виде невозможно ввиду большого количества вершин в графе Γ. В качестве альтернативы коммутации вида (1) рассмотрим коммутацию

$$(i_0, i_1, i_2) \xrightarrow{\tau} (i_0, i_1 + i_0 \bmod m', i_2 + i_0 \bmod m'). \tag{9}$$

При $m' = 3$ коммутация имеет следующий вид:

$$1, 14, 27, 4, 17, 21, 7, 11, 24, 10, 23, 9, 13, 26, 3, 1620, 6, 19, 5, 18, 22, 8, 12, 25, 2, 15.$$

С одной стороны, данная коммутация обеспечивает указанный принцип: в графе Γ в каждую вершину заходит и выходит по m' рёбер в разные вершины соответствующих слоёв. Однако эта коммутация сохраняет слабости коммутации вида (1). На это обстоятельство автору указал Ф. М. Малышев. Таким образом, имеет место следующее

Утверждение 1. Пусть рассматриваемый XSLP-алгоритм имеет длину блока $n = n'(m')^3$ и преобразование P задаётся коммутацией вида (9). Тогда для показателя рассеивания $\theta = \theta(r)$ линейной среды XSLP-алгоритма на $r \in \mathbb{N}$ итераций выполнено:

$$\theta(r) = \begin{cases} 1, & \text{если } r = 1, \\ \rho, & \text{если } r = 2, \\ 2\rho - 1, & \text{если } r = 3, \\ \rho^2, & \text{если } r = 4, \\ \theta(r-4) + \rho^2, & \text{если } r > 4. \end{cases}$$

Доказательство. Для доказательства утверждения нужно лишь заметить, что в графе Γ существует подграф Γ_1, который соответствует XSLP-алгоритму с длиной блока $n_1 = n' \cdot m^2$ и равномерно рассеивающими перестановками (в терминологии [20]), то есть коммутацией вида (1).

Действительно, выделим одну вершину в графе Γ с номером $i \in \{1, \ldots, (m')^2\}$. Данная вершина инцидентна рёбрам, которые соответствуют подстановкам с номерами $(i-1)m'+1$, $(i-1)m'+2$, …, $(i-1)m'+m'$. Представив эти номера в виде $i_0 + m'j_1 + (m')^2 j_2$, получаем, что коммутация τ действует следующим образом:

$$(i_0, i_1, i_2) \xrightarrow{\tau} (i_0, i_1 + i_0 \bmod m', i_2 + i_0 \bmod m') \xrightarrow{\tau} (i_0, i_1, i_2).$$

Это с точностью до перенумерации соответствует коммутации вида (1). Таким образом, в каждом слое графа Γ_1 всего m' вершин, в каждую из которой входит не более m' рёбер и выходит не более m' рёбер, причём это справедливо при любой выборе $i \in \{1, \ldots, (m')^2\}$.

Осталось заметить, что для графа Γ_1, согласно [20], выполнено

$$\upsilon(\Gamma^{(1)}) = 1,\ \upsilon(\Gamma^{(2)}) = \rho,\ \upsilon(\Gamma^{(3)}) = 2\rho - 1,\ \upsilon(\Gamma^{(4)}) = \rho^2,\ \upsilon(\Gamma^{(5)}) = \upsilon(\Gamma^{(r-4)}) + \rho^2,$$

что завершает доказательство утверждения. ■

В качестве примера, демонстрирующего результат утверждения 1, рассмотрим случай $m' = 3$. Для получаемого в этом случае XSLP-алгоритма подграф Γ_1 графа Γ приведён на рис. 6.

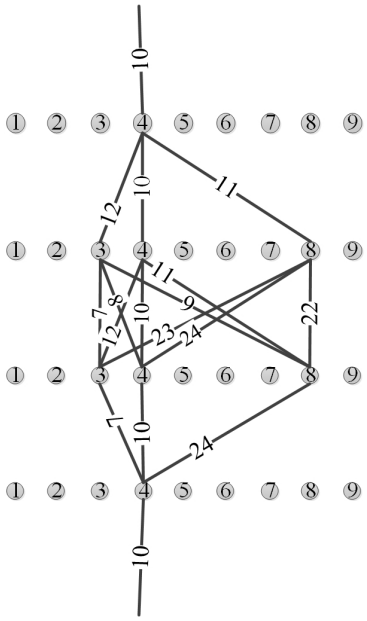

Рис. 6. Подграф Γ_1 графа Γ XSLP-алгоритма с длиной блока $n = n'(m')^3$ при $m' = 3$ и коммутации вида (9). Рёбра и вершины занумерованы слева направо, начиная с 1

ЛИТЕРАТУРА

1. *Трифонов Д. И., Фомин Д. Б.* Об инвариантных подпространствах в XSL-шифрах // Прикладная дискретная математика. 2021. Т. 54. С. 59–77.
2. ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры. М.: Стандартинформ, 2015.
3. *Burov D. A. and Pogorelov B. A.* An attack on 6 rounds of KHAZAD // Матем. вопр. криптогр. 2016. Т. 7. № 2. С. 35–46.
4. *Глухов М. М., Елизаров В. П., Нечаев А. А.* Алгебра. В 2-х т. Т. 1. М.: Гелиос АРВ, 2003. 336 с.
5. *Глухов М. М., Елизаров В. П., Нечаев А. А.* Алгебра. В 2-х т. Т. 2. М.: Гелиос АРВ, 2003. 416 с.
6. *Малышев Ф. М., Тараканов В. Е.* Обобщенные графы де Брейна // Матем. заметки. 1997. Т. 62. Вып. 4. С. 540–548.
7. *Daemon J. and Rijmen V.* The Design of Rijndael: AES — The Advanced Encryption Standard. Berlin; Heidelberg: Springer, 2002. 238 p.
8. *Ерохин А. В., Малышев Ф. М., Тришин А. Е.* Многомерный линейный метод и показатели рассеивания линейной среды шифрпреобразований // Матем. вопр. криптогр. 2017. Т. 8. № 4. С. 29–62.
9. Advanced Encryption Standard (AES). `https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf`. 2001.
10. ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования. М.: Стандартинформ, 2012.
11. *Brown L., Kwan M., Pieprzyk J., and Seberry J.* Improving resistance to differential cryptanalysis and the redesign of LOKI // LNCS. 1993. V. 739. P. 36–50.
12. *Knudsen L. R.* Cryptanalysis of LOKI // LNCS. 1993. V. 739. P. 22–35.
13. Data Encryption Standard (DES). NIST FIPS PUB 46. 1977.

14. *Massey J. L.* SAFER K-64: A byte-oriented block-ciphering algorithm // LNCS. 1994. V. 809. P. 1–17.
15. *Massey J. L.* SAFER K-64: One year later // LNCS. 1995. V. 1008. P. 212–241.
16. *Feistel H.* Cryptography and computer privacy // Scientific Amer. 1973. V. 228. No. 5. P. 15–23.
17. *Малышев Ф. М.* Двойственность разностного и линейного методов в криптографии // Матем. вопр. криптогр. 2014. Т. 5. № 3. С. 35–48.
18. *Malyshev F. M. and Trishin A. E.* Linear and differential cryptanalysis: Another viewpoint // Матем. вопр. криптогр. 2020. Т. 11. № 2. С. 83–98.
19. *Малышев Ф. М., Трифонов Д. И.* Рассеивающие свойства XSLP-шифров // Матем. вопр. криптогр. 2016. Т. 7. № 3. С. 47–60.
20. *Федченко В. А.* Показатели рассеивания линейной среды AES-подобных алгоритмов шифрования // Матем. вопр. криптогр. 2017. Т. 8. № 3. С. 109–126.

REFERENCES

1. *Trifonov D. I. and Fomin D. B.* Ob invariantnykh podprostranstvakh v XSL-shifrakh [Invariant subspaces in SPN block c ipher]. Prikladnaya Diskretnaya Matematika, 2021, no. 54, pp. 59–77. (in Russian)
2. GOST R 34.12-2015. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Blochnyye shifry. [Information Technology. Cryptographic Information Protection. Block ciphers]. Moscow, Standartinform, 2015. (in Russian)
3. *Burov D. A. and Pogorelov B. A.* An attack on 6 rounds of KHAZAD. Matem. Vopr. Kriptogr., 2016, vol. 7, no. 2, pp. 35–46.
4. *Gluhov M. M., Elizarov V. P., and Nechaev A. A.* Algebra [Algebra]. Vol. 1. Moscow, Gelios ARV Publ., 2003. 336 p. (in Russian)
5. *Gluhov M. M., Elizarov V. P., and Nechaev A. A.* Algebra [Algebra]. Vol. 2. Moscow, Gelios ARV Publ., 2003. 416 p. (in Russian)
6. *Malyshev F. M. and Tarakanov V. E.* Obobshchennyye grafy de Breyna [Generalized de Bruijn graphs]. Matem. Zametki, 1997, vol. 62, no. 4, pp. 540–548. (in Russian)
7. *Daemon J. and Rijmen V.* The Design of Rijndael: AES — The Advanced Encryption Standard. Berlin; Heidelberg, Springer, 2002, 238 p.
8. *Erokhin A. V., Malyshev F. M., and Trishin A. E.* Mnogomernyy lineynyy metod i pokazateli rasseivaniya lineynoy sredy shifrpreobrazovaniy [Multidimensional linear method and diffusion characteristics of linear medium of ciphering transform]. Matem. Vopr. Kriptogr., 2017, vol. 8, no. 4, pp. 29–62. (in Russian)
9. Advanced Encryption Standard (AES). `https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf`, 2001.
10. GOST R 34.11-2012. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Funktsiya kheshirovaniya. [Information Technology. Cryptographic Information Protection. Hash Function]. Moscow, Standartinform, 2012. (in Russian)
11. *Brown L., Kwan M., Pieprzyk J., and Seberry J.* Improving resistance to differential cryptanalysis and the redesign of LOKI. LNCS, 1993, vol. 739, pp. 36–50.
12. *Knudsen L. R.* Cryptanalysis of LOKI. LNCS, 1993, vol. 739, pp. 22–35.
13. Data Encryption Standard (DES). NIST FIPS PUB 46. 1977.
14. *Massey J. L.* SAFER K-64: A byte-oriented block-ciphering algorithm. LNCS, 1994, vol. 809, pp. 1–17.
15. *Massey J. L.* SAFER K-64: One year later. LNCS, 1995, vol. 1008, pp. 212–241.

16. *Feistel H.* Cryptography and computer privacy. Scientific Amer., 1973, vol. 228, no. 5, pp. 15–23.
17. *Malyshev F. M.* Dvojstvennost' raznostnogo i linejnogo metodov v kriptografii [The duality of differential and linear methods in cryptography]. Matem. Vopr. Kriptogr., 2014, vol. 5, no. 3, pp. 35–48. (in Russian)
18. *Malyshev F. M. and Trishin A. E.* Linear and differential cryptanalysis: Another viewpoint. Matem. Vopr. Kriptogr., 2020, vol. 11, no. 2, pp. 83–98.
19. *Malyshev F. M. and Trifonov D. I.* Rasseivayushchiye svoystva XSLP-shifrov [Diffusion properties of XSLP-ciphers]. Matem. Vopr. Kriptogr., 2016, vol. 7, no. 3, pp. 47–60. (in Russian)
20. *Fedchenko V. A.* Pokazateli rasseivaniya lineynoy sredy AES-podobnykh algoritmov shifrovaniya [Diffusion rates of linear medium in AES-like ciphers]. Matem. Vopr. Kriptogr., 2017, vol. 8, no. 3, pp. 109–126. (in Russian)