УДК 519.1+519.719.2+519.712

КРИПТОСИСТЕМА ШИФРОВАНИЯ С АУТЕНТИФИКАЦИЕЙ С ПРОИЗВОДНЫМИ РАЗОВЫМИ КЛЮЧАМИ

А. Ю. Зубов

ООО «Центр сертификационных исследований», г. Москва, Россия

Продолжается исследование предложенной автором математической модели криптосистемы шифрования с аутентификацией на основе кода аутентификации с секретностью. Алгоритм шифрования использует вычисления в полях характеристики два, счётчиковую последовательность, зависящую от ключа, одноразовые производные ключи, определяемые основным ключом и вектором инициализации с помощью ортогональных латинских квадратов, а также имитовставку полиномиального типа. Предлагается байтовый метод реализации алгоритма, проводится его сравнение со стандартизованным криптоалгоритмом GCM. Предлагается выбор параметров модели, гарантирующих доказуемую стойкость к атакам на основе шифртекста. Проводится анализ стойкости к атакам с выбранным открытым текстом.

Ключевые слова: *криптосистема шифрования с аутентификацией, GCM, квазигруппа, ортогональные латинские квадраты, доказуемая стойкость, атаки на основе шифртекста, атаки с выбранным открытым текстом.*

DOI 10.17223/20710410/50/3

AUTHENTICATION ENCRYPTION CRYPTOSYSTEM WITH DERIVED ONE-TIME KEYS

A. Yu. Zubov

Certification Research Center, Moscow, Russia

E-mail: zubovanatoty@yandex.ru

A research of the previously proposed by the author mathematical model of authenticated encryption cryptosystem based on authentication code with secrecy is continued. An encryption algorithm uses calculations in the fields of characteristics two, a counters sequence depending on the key, one-time derived keys defined by the main key and initialization vectors using orthogonal Latin squares, and a polynomial-type MAC. A byte method for implementing the algorithm is proposed and compared with the standardized GCM cryptographic algorithm. The choice of model parameters that guarantee provable security to ciphertext-based attacks is proposed. The analysis of the cryptosystem's security to chosen-plaintext attacks is performed.

Keywords: *authenticated encryption cryptosystem, GCM, quasigroup, orthogonal Latin squares, provable security, ciphertext-based attacks, chosen-plaintext attacks.*

1. Базовая конструкция криптосистемы $\mathbb{CS}$

В [1] предложена модель криптосистемы $\mathbb{CS}$ шифрования с аутентификацией на основе кода аутентификации с секретностью. Настоящая работа продолжает начатые в [1] исследования этой модели. Приведём её описание.

Будем использовать обозначения a, $\overline{a}$, $\mathbf{a}$ для элемента поля $a \in \mathbb{F}_{2^n} = \mathrm{GF}(2^n)$, строки $a \in \{0,1\}^n$ коэффициентов многочлена, представляющего a как элемент факторкольца $\mathbb{F}_2[x]/(h(x)) \cong \mathbb{F}_{2^n}$, и числа $\mathbf{a} \in \mathbb{Z}_{2^n}$, двоичная запись которого совпадает с $\overline{a}$.

Криптосистема $\mathbb{CS}$ использует для зашифрования открытых текстов $s^{(\tau)}$, $\tau = 1, 2, \ldots, N$, секретный ключ

$$k = (a, b) \in (\mathbb{F}_{2^n})^2,$$

векторы инициализации

$$iv^{(\tau)} = \left(\gamma^{(\tau)}, \delta^{(\tau)}\right) \in (\mathbb{F}_{2^n})^2,$$

отображения

$$f_x : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n},\ x \in (\mathbb{F}_{2^n})^2, \quad \varphi : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}, \quad g : (\mathbb{F}_{2^n})^4 \to (\mathbb{F}_{2^n})^2$$

и наборы констант

$$\{w_i \in \mathbb{F}_{2^n} : \mathbf{w}_i = (i + i_0) \pmod{2^n}\}, \quad \{c_i(k), d_i(k)\},$$
$$c_i(k) = f_k(w_i), \quad d_i(k) = \varphi(c_i(k)), \quad i = 1, 2, \ldots,$$

где $i_0 \geqslant 0$ — число, определяющее начальное значение счётчиковой последовательности $\{w_i\}$.

Вектор инициализации должен содержать счётчик, который делает вектор неповторяющимся, метку времени и атрибуты передаваемого сообщения. Текст $s^{(\tau)}$, представленный строкой

$$s^{(\tau)} = \left(s_1^{(\tau)}, \ldots, s_{r_\tau}^{(\tau)}\right)$$

элементов из $\mathbb{F}_{2^n}$, шифруется на производном ключе

$$k^{(\tau)} = \left(a^{(\tau)}, b^{(\tau)}\right) = g\left(k, iv^{(\tau)}\right) \in (\mathbb{F}_{2^n})^2.$$

Результат зашифрования $m^{(\tau)} = \left(m_1^{(\tau)}, \ldots, m_{r_\tau}^{(\tau)}, m_{r_\tau+1}^{(\tau)}\right)$ — это строка элементов из $\mathbb{F}_{2^n}$, где

$$m_i^{(\tau)} = s_i^{(\tau)} \oplus c_i(k) a^{(\tau)} \oplus d_i(k) b^{(\tau)}, \quad i = 1, 2, \ldots, r_\tau,\ r_\tau > 1; \tag{1}$$

$$m_{r_\tau+1}^{(\tau)} = a^{(\tau)} \oplus m_1^{(\tau)} \left(b^{(\tau)}\right)^{r_\tau} \oplus \ldots \oplus m_{r_\tau}^{(\tau)} \left(b^{(\tau)}\right)^1. \tag{2}$$

Отправитель посылает сообщение (iv, m). Получатель сообщения (iv, m), где $m = (m_1, \ldots, m_r, m_{r+1})$, вычисляет производный ключ

$$k' = (a', b') = g(k, iv).$$

Критерий аутентичности сообщения — равенство

$$m_{r+1} = a' \oplus m_1 \cdot b'^r \oplus \ldots \oplus m_r \cdot b'^1.$$

Если оно выполнено, то получатель вычисляет $c_i(k), d_i(k)$ и текст $s = (s_1, \ldots, s_r)$, где

$$s_i = m_i \oplus c_i(k) a' \oplus d_i(k) b', \quad i = 1, 2, \ldots, r.$$

При случайном выборе ключа k не исключается случай, когда $k^{(\tau)} = (0, 0)$. Он нежелателен, поскольку все блоки гаммы становятся равными 0^n. Избавиться от этого

недостатка можно следующим образом. При вычислении ключа $k^{(\tau)}$ вводится дополнительный шаг — проверка равенства $k^{(\tau)} = (0, 0)$. Если оно выполняется, то нужно сменить $iv^{(\tau)}$. Для этого, например, в $\overline{iv^{(\tau)}}$ выделить один бит и в случае, когда равенство выполняется, инвертировать этот бит.

На рис. 1 представлена упрощённая схема алгоритма $\mathbb{CS}$ (без использования ассоциированных данных и дополнения последнего неполного блока).

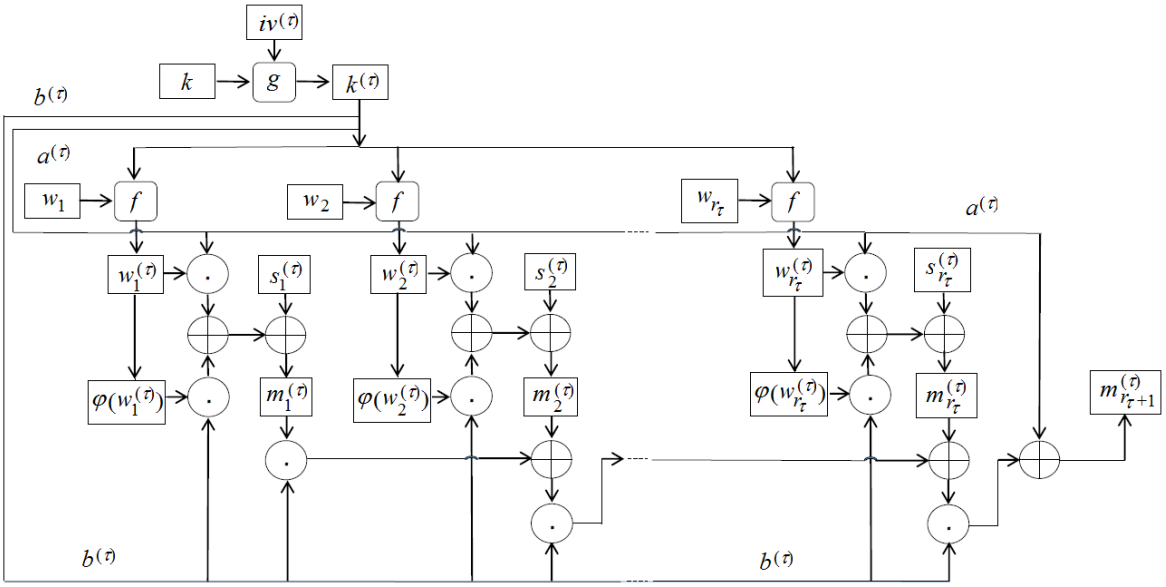

Рис. 1. Схема алгоритма $\mathbb{CS}$

Выбор отображений φ, f_k, g диктуется следующими условиями:

a) f_k инъективно;
b) $c_i(k)d_j(k) \neq c_j(k)d_i(k)$ при $i \neq j$;
c) разным τ отвечают разные $k^{(\tau)}$;
d) при случайном равновероятном выборе k случайно равновероятно выбирается и $k^{(\tau)}$.

Требования (a)–(d) необходимы для получения оценок стойкости криптосистемы к атакам на основе шифртекста (см. далее п. 5).

2. Выбор отображений

В [1] предложен следующий выбор отображений φ, f_k, g для $k = (a, b)$ и $iv = (\gamma, \delta)$:

$$\varphi(\vartheta) = \vartheta^2 \text{ либо } \varphi(\vartheta) = \vartheta \oplus \alpha,$$

где α — любой ненулевой элемент из $\mathbb{F}_{2^n}$;

$$f_k(w) = \Lambda\left(w * a\right) *' b; \quad g\left(k, iv\right) = \left(\Lambda(a * \gamma) *' a, \Lambda(b * \delta) *' b\right), \tag{3}$$

в которых используются квазигрупповые операции $*$, $*'$, определяемые значениями подходящих многочленов $P(x, y) \in \mathbb{Z}_{2^n}[x, y]$, и биективное преобразование $\Lambda : \mathbb{F}_{2^n} \to \to \mathbb{F}_{2^n}$.

Анализ выбора параметров алгоритма в варианте (3) обнаружил его недостатки. Так, оказалось, что отображение g, определённое формулой (3), может не гарантировать свойство (d), поскольку не всякие квазигрупповые операции указанного типа

обеспечивают равномерное распределение на множестве производных ключей. Кроме того, производные ключи достаточно сложно вычисляются. В связи с этим предлагается новый вариант, устраняющий указанные недостатки.

Определим φ и Λ так же, как в (3), и положим

$$f_k(w) = \Lambda\left(w * a\right) * b, \;\; g\left(k, iv\right) = \left(a *_{iv} b, \;\; a *'_{iv} b\right), \tag{4}$$

где $k = (a, b)$; $iv = (\gamma, \delta)$; $*_{iv}$, $*'_{iv}$ — операции на $\mathbb{F}_{2^n}$, определяемые ортогональными латинскими квадратами L_1, L_2 на $\mathbb{F}_{2^n}$ следующим образом.

Пусть L_1, L_2 — таблицы Кэли квазигрупп $(\mathbb{F}_{2^n}, *)$, $(\mathbb{F}_{2^n}, *')$ соответственно; $L_\alpha(x, y)$ — элемент латинского квадрата L_α, расположенный в клетке (x, y). Тогда положим

$$a^{(\tau)} = a *_{iv} b = L_1\left(\Lambda\left(a * \gamma\right), \Lambda\left(b * \delta\right)\right), \;\; b^{(\tau)} = a *'_{iv} b = L_2\left(\Lambda\left(a * \gamma\right), \Lambda\left(b * \delta\right)\right). \tag{5}$$

В (4) предлагается использовать ту же операцию $*$, что и в (5).

Несложно видеть, что для любого вектора инициализации iv таблицы Кэли операций $*_{iv}$ и $*'_{iv}$, определённых формулами (5), являются ортогональными латинскими квадратами на $\mathbb{F}_{2^n}$. Это следует из того, что L_1, L_2 — ортогональные латинские квадраты и Λ — биективное преобразование поля $\mathbb{F}_{2^n}$. Далее мы уточним выбор L_1, L_2.

Заметим, что вариант (4) выбора отображений гарантирует выполнение свойств (a)–(d). Свойства (a), (b) обеспечиваются выбором отображений φ и f_k. Пусть $*_{iv}$,$*'_{iv}$ — операции, определённые формулами (5). Тогда при любых $u, v \in \mathbb{F}_{2^n}$ система уравнений

$$\begin{cases} a *_{iv} b = u, \\ a *'_{iv} b = v \end{cases} \tag{6}$$

имеет единственное решение (a, b) для любого iv. Отсюда следует свойство (d). Из формул (5) также очевидно, что разным векторам инициализации $iv^{(\tau)}$ и $iv^{(\tau')}$ отвечают разные производные ключи $k^{(\tau)}$ и $k^{(\tau')}$, поэтому выполняется условие (c).

3. Байтовый метод

Рассмотрим следующий вариант квазигрупповых операций в формулах (4) и (5).

Пусть x, y — элементы поля $\mathbb{F}_{2^n}$ и $\overline{x}$, $\overline{y}$ — соответствующие им векторы из $\{0,1\}^n$. Представим $\overline{x}, \overline{y}$ в виде последовательности байтов $\overline{x} = \overline{x}_1 || \ldots || \overline{x}_{n/8}$, $\overline{y} = \overline{y}_1 || \ldots || \overline{y}_{n/8}$. Будем интерпретировать байты $\overline{x}_i$ и $\overline{y}_i$ как элементы x_i, y_i поля $\mathbb{F}_{2^8} = \mathrm{GF}(2)/(x^8 + x^7 + x^6 + x + 1)$ и использовать запись $x = \left(x_1, \ldots, x_{n/8}\right)$, $y = \left(y_1, \ldots, y_{n/8}\right)$.

Пусть $\overline{*}$ — квазигрупповая операция на $\mathbb{F}_{2^8}$. Определим на $\mathbb{F}_{2^n}$ операцию $*$ как

$$x * y = \left(x_1 \,\overline{*}\, y_1, \ldots, x_{n/8} \,\overline{*}\, y_{n/8}\right). \tag{7}$$

Ясно, что $*$ является квазигрупповой операцией на $\mathbb{F}_{2^n}$.

В (4) произведение $w_i * a$ будем вычислять в соответствии с (7), а $f_k(w)$ — следующим образом. Пусть $\overline{\Lambda}$ — матрица размера $n/8 \times n/8$ над $\mathbb{F}_{2^8}$, обладающая хорошими рассеивающими свойствами. Например, при $n = 128$ в качестве $\overline{\Lambda}$ можно выбрать максимально рассеивающую матрицу, используемую в шифрсистеме «Кузнечик». Пусть $y = w_i * a = \left(y_1, \ldots, y_{n/8}\right)$ и

$$z = (z_1, \ldots, z_8) = \left(y_1, \ldots, y_{n/8}\right) \overline{\Lambda}. \tag{8}$$

Тогда $f_k(w) = z * b$ будем вычислять в соответствии с формулой (7).

В описании байтового метода остаётся объяснить выбор квазигрупповой операции $\overline{*}$.

Построим пару ортогональных латинских квадратов M_1, M_2 на $\mathbb{F}_{2^8}$, используя, например, метод Боуза [2], который состоит в следующем: пусть

$$\mathbb{F}_{2^t} = \{\alpha_0 = 0, \alpha_1 = 1, \alpha_2, \ldots, \alpha_{2^t-1}\}.$$

Тогда матрицы

$$M_r = \left(\alpha_{i,j}^{(r)}\right), \quad r = 1, \ldots, 2^t - 1,$$

где

$$\alpha_{i,j}^{(r)} = \alpha_i \cdot \alpha_r \oplus \alpha_j, \quad i, j = 0, 1, \ldots, 2^t - 1,$$

образуют полную систему ортогональных латинских квадратов на $\mathbb{F}_{2^t}$. Произвольную пару таких квадратов при $t = 8$ выберем в качестве M_1, M_2. Вместе с ними можем получить порядка $(256!)^3$ других пар $M_1^{(\pi)}, M_2^{(\pi)}$ ортогональных латинских квадратов путём одинаковых перестановок в M_1, M_2 строк, столбцов и перенумераций элементов. Каждая из пар $M_1^{(\pi)}, M_2^{(\pi)}$ определяет квазигрупповые операции $\overline{*}$ и $\overline{*}'$ на $\mathbb{F}_{2^8}$.

Недостатком латинских квадратов M_1, M_2, построенных методом Боуза, является достаточно простое алгебраическое задание соответствующих операций $\overline{*}$ и $\overline{*}'$, позволяющее аналитическое исследование связи между основным и производным ключами. Поэтому предлагается выбрать пару латинских квадратов $M_1^{(\pi)}, M_2^{(\pi)}$, для которых операции $\overline{*}$ и $\overline{*}'$ не будут обладать «хорошими» алгебраическими свойствами, например свойствами коммутативности, ассоциативности, дистрибутивности относительно $\oplus$ и т. д. Это сделать нетрудно. Для наших целей целесообразно также выбирать такую операцию $\overline{*}$, чтобы функция $F : (\mathbb{F}_{2^8})^2 \to \mathbb{F}_{2^8}$, определённая формулой $F(u, v) = u\overline{*}v$, имела как можно большую нелинейность в том смысле, как это принято в теории дискретных функций. Её выбор представляет предмет отдельного исследования. Отметим, что число возможных пар $M_1^{(\pi)}, M_2^{(\pi)}$ делает практически невозможной попытку восстановления исходных латинских квадратов M_1, M_2 по выбранным латинским квадратам $M_1^{(\pi)}, M_2^{(\pi)}$.

После того как сделан выбор $M_1^{(\pi)}, M_2^{(\pi)}$, используем соответствующие им операции $\overline{*}$ и $\overline{*}'$ для определения операций $*, *'$ по формуле (7). В качестве L_1, L_2 в формуле (5) будем использовать таблицы Кэли операций $*, *'$. Для хранения и использования выбранных квадратов потребуется память объёма $2 \cdot 256^2$ байтов.

Недостатком байтового метода является необходимость использования достаточно большого объёма памяти в 131072 байта. Значительно меньший объём памяти потребуется для «полубайтового» метода, который строится аналогично байтовому путём замены байтов на полубайты. В этом случае объём памяти уменьшается до 512 байтов.

4. Анализ эффективности реализации алгоритма $\mathbb{CS}$

Преимущество байтового (полубайтового) метода состоит в том, что вычисления в $\mathbb{F}_{2^8}$ (соответственно в $\mathbb{F}_{2^4}$) реализуются значительно проще, чем в $\mathbb{F}_{2^n}$ при $n \geqslant 64$. Кроме того, необходимые операции задаются таблично. Например, при $n = 128$ для этого потребуется память объёма 131328 байтов для байтового и 768 байтов для полубайтового методов для записи двух латинских квадратов $M_1^{(\pi)}, M_2^{(\pi)}$ и матрицы $\overline{\Lambda}$. Обращения в такую память при выполнении операций алгоритма сравнимы по сложности с булевыми операциями. Единственными «сложными» операциями алгоритма $\mathbb{CS}$ в таком случае остаются операции умножения в поле $\mathbb{F}_{2^n}$ в формулах (1) и (2). Число

операций умножения в поле $\mathbb{F}_{2^n}$, выполняемых алгоритмом $\mathbb{CS}$, сопоставимо с числом аналогичных операций стандартизованного алгоритма GCM [3, 4].

Чтобы убедиться в этом, приведём схему алгоритма GCM (рис. 2) и оценим сложность двух алгоритмов в сопоставимых операциях, пользуясь при реализации $\mathbb{CS}$ формулами (5) в качестве метода вычисления производного ключа и байтовым методом.

Рассмотрим лишь упрощённые схемы алгоритмов (не использующие ассоциированные данные), полагая, что открытые тексты состоят из полных n-битовых блоков.

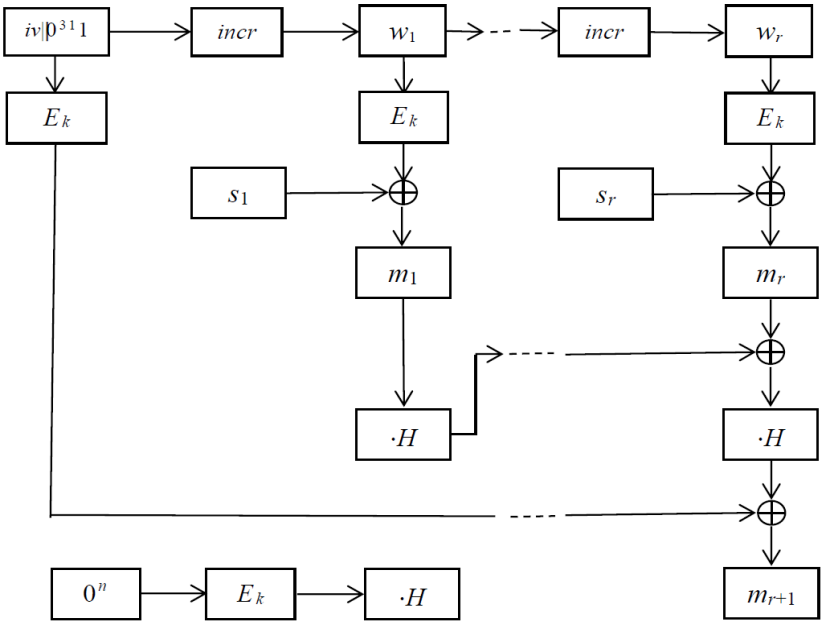

Рис. 2. Схема алгоритма GCM

Пусть $T_g, T_f, T_\oplus, T_\odot, T_\ll, T_P$ — соответственно трудоёмкости выполнения операций $g_y, f_x, \oplus, \odot, \ll$ и обращения в память, где $\oplus$ — операция XOR; $\odot$ — операция умножения в поле $\mathbb{F}_{2^n}$; $\ll$ — операция сдвига двоичного вектора.

В случае, когда $\varphi(\vartheta) = \vartheta^2$, трудоёмкость $T_{\mathbb{CS}}$ алгоритма зашифрования сообщения s, состоящего из r блоков, выражается формулой

$$T_{\mathbb{CS}} = [T_g + rT_f + 2rT_\oplus + 3rT_\odot] + r\,(T_\oplus + T_\odot)\,, \tag{9}$$

где слагаемое в квадратных скобках — трудоёмкость получения r блоков шифртекста, а второе слагаемое — трудоёмкость получения имитовставки.

Из схемы алгоритма GCM находим аналогичную величину для T_{GCM}:

$$T_{\mathrm{GCM}} = [(r+2)\,T_E + rT_\oplus] + r\,(T_\oplus + T_\odot)\,. \tag{10}$$

Здесь T_E — трудоёмкость зашифрования одного блока данных для GCM.

Для сравнения $T_{\mathbb{CS}}$ и T_{GCM} сделаем грубую прикидку. Сошлёмся на работы [3, 4], в которых указывается, что в случае, когда GCM использует для шифрования AES, имеет место соотношение $T_E \approx 3T_\odot$. В свою очередь будем полагать, что операция $\odot$ равносильна $3n$ операциям типа $\ll$ и $\oplus$. Это можно пояснить следующим образом.

Пусть, например, $\mathbb{F}_{2^{128}}$ порождается многочленом $x^{128}+x^7+x^2+x+1$. Тогда для $\overline{\alpha}=(\alpha_{127},\ldots,\alpha_1,\alpha_0)$ произведение $\alpha\cdot x$ можно вычислить, пользуясь формулой

$$\overline{\alpha\cdot x}=\begin{cases}\overline{\alpha}\ll 1, & \text{если } \alpha_{127}=0,\\ (\overline{\alpha}\ll 1)\oplus 0^{120}||10000111, & \text{если } \alpha_{127}=1.\end{cases}$$

Представив $\overline{\beta}=(\beta_{127},\ldots,\beta_1,\beta_0)$ в виде многочлена $\beta_{127}x^{127}+\ldots+\beta_1x+\beta_0$, вычислим произведение $\alpha\cdot\beta$ не более чем за $3\cdot 128$ операций указанного типа.

Заметим, что произведение (8) вычисляется за $n/8$ операций обращения в память и $n/8-1$ операций XOR. С учётом этого из (4), (5), (7), (8) получаем

$$T_g=(n-4)T_P+(n/4)T_{\oplus}; \tag{11}$$

$$T_f=(3n/8)\,T_P+(n/8-1)\,T_{\oplus}. \tag{12}$$

Будем также полагать, что $T_{\oplus},T_{\ll},T_P$ равносильны (по сложности или времени выполнения). Поскольку основные операции алгоритмов выражаются через эти операции, будем каждую из них считать элементарной. Выразим $T_{\mathbb{CS}}$ и T_{GCM} в элементарных операциях. Непосредственно из (9)–(12) получаем

$$T_{\mathbb{CS}}=12{,}5rn+1{,}25n+2r-5,\quad T_{\mathrm{GCM}}=12rn+18n+2r,$$

или приближённо (с учётом того, что r достаточно велико)

$$T_{\mathbb{CS}}\approx 12{,}5rn,\quad T_{\mathrm{GCM}}\approx 12rn.$$

Отметим возможность повышения эффективности алгоритма $\mathbb{CS}$ за счёт использования дополнительной памяти для хранения констант $c_i(k),d_i(k)$.

Пусть необходимо зашифровать ряд сообщений длины r (где r не слишком велико) с помощью основного ключа k. Будем записывать вычисленные при зашифровании первого сообщения константы $c_i(k),d_i(k)$, $i=1,2,\ldots,r$, в память объёма $2rn$ битов, а при шифровании второго и следующего сообщений обращаться в эту память, извлекая из неё необходимую информацию. Такой способ позволит сэкономить за счёт того, что обращение в память значительно проще, чем вычисление констант $c_i(k),d_i(k)$.

Аналогичный расчёт для случая, когда в алгоритме $\mathbb{CS}$ используется отображение $\varphi(\vartheta)=\vartheta+\alpha$, $\alpha\in\mathbb{F}_{2^n}$, даёт следующие результаты:

$$T_{\mathbb{CS}}=9{,}5rn,\quad T_{\mathrm{GCM}}=12rn.$$

Конечно, при сравнении величин $T_{\mathbb{CS}}$ и T_{GCM} необходимо учитывать поправку на то, что при реализации $\mathbb{CS}$ операция умножения $\odot$ «сто́ит» несколько «дороже», чем при реализации GCM. Дело в том, что при шифровании каждого сообщения с помощью $\mathbb{CS}$ приходится вычислять произведения элементов поля $\mathbb{F}_{2^n}$ на компоненты $a^{(\tau)},b^{(\tau)}$ очередного производного ключа $k^{(\tau)}$, а для GCM — лишь произведения элементов поля на один и тот же ключ аутентификации $H=E_k(\overline{0})$.

Более точное сравнение алгоритмов возможно лишь для конкретных реализаций.

5. Анализ стойкости криптосистемы $\mathbb{CS}$

Рассмотрим возможность определения ключа $k=(a,b)$ криптосистемы $\mathbb{CS}$ в атаке с выбранным открытым текстом. Ключ k находится из системы уравнений

$$\begin{cases}c_i(k)\cdot a^{(\tau)}\oplus d_i(k)b^{(\tau)}=u_i^{(\tau)},\\ a^{(\tau)}\oplus m_1^{(\tau)}\left(b^{(\tau)}\right)^{r_\tau}\oplus\ldots\oplus m_{r_\tau}^{(\tau)}\left(b^{(\tau)}\right)^{1}=m_{r_\tau+1}^{(\tau)},\\ i=1,2,\ldots,r_\tau,\ \tau=1,2,\ldots,L,\end{cases} \tag{13}$$

в которой $u_i^{(\tau)} = s_i^{(\tau)} \oplus m_i^{(\tau)}$, $a^{(\tau)}$, $b^{(\tau)}$ определяются формулами (5), $c_i(k) = \Lambda\,(w * a) * b$, $d_i(k) = (c_i(k))^2$ либо $d_i(k) = c_i(k) \oplus \alpha$. Несмотря на то, что эта система содержит много уравнений и всего две переменные (a и b), её аналитическое решение не представляется возможным в силу разнородности используемых операций.

Вид уравнений системы (13) приводит к естественной задаче нахождения из этой системы хотя бы одного производного ключа. Если эта задача будет успешно решена, то это облегчит нахождение основного ключа. Рассмотрим этот подход.

Находить производный ключ $k^{(\tau)} = \left(a^{(\tau)}, b^{(\tau)}\right)$ можно перебором возможных значений $b^{(\tau)}$, вычислением из второго уравнения соответствующего $a^{(\tau)}$ и проверкой получившегося допустимого варианта $k^{(\tau)}$ с помощью первого уравнения. Сложность этого вычисления оценивается величиной $O(2^n)$. Проверка допустимого варианта невозможна, если неизвестны $c_i(k)$. Чтобы найти их, требуется ключ k. Но если при этом, имея допустимый вариант $k^{(\tau)}$, удастся найти k с помощью метода, основанного на взаимной зависимости констант $c_i(k), c_j(k)$, с трудоёмкостью меньшей $O(2^n)$, то можно получить метод определения ключа со сложностью меньше полного перебора ключей. Покажем, что использование в формуле $c_i(k) = \Lambda\,(w * a) * b$ рассеивающего преобразования Λ устраняет указанную возможность при использовании байтового метода.

Пусть константа $c_i(k)$ определяется формулой $c_i(k) = (w * a) * b$. Тогда заметим, что если векторы $\overline{w}_i$ и $\overline{w}_{i+1}$ отличаются лишь одним байтом, то и векторы $\overline{f_k\,(w_i)}, \overline{f_k\,(w_{i+1})}$ также отличаются лишь одним байтом, т. е. сумма $f_k\,(w_i) \oplus f_k\,(w_{i+1}) = w$ принадлежит известному множеству, мощность которого не превосходит 2^8. В случае, когда $\varphi(\vartheta) = \vartheta^2$, из первого уравнения системы (13) получаем равенства

$$f_k\,(w_i)\,a^{(\tau)} \oplus f_k^2\,(w_i)\,b^{(\tau)} = u_i^{(\tau)},$$
$$f_k\,(w_{i+1})\,a^{(\tau)} \oplus f_k^2\,(w_{i+1})\,b^{(\tau)} = u_{i+1}^{(\tau)},$$

складывая которые, получаем $w \cdot a^{(\tau)} \oplus w^2 \cdot b^{(\tau)} = u$, где $u = u_i^{(\tau)} \oplus u_{i+1}^{(\tau)}$. Аналогично, при $j \neq i$ получаем $w' \cdot a^{(\tau)} \oplus w'^2 \cdot b^{(\tau)} = u'$, где $u' = u_j^{(\tau)} \oplus u_{j+1}^{(\tau)}$, а w' принадлежит известному множеству, мощность которого не превосходит 2^8. В результате имеем систему уравнений

$$\begin{cases} w \cdot a^{(\tau)} \oplus w^2 \cdot b^{(\tau)} = u, \\ w' \cdot a^{(\tau)} \oplus w'^2 \cdot b^{(\tau)} = u' \end{cases}$$

относительно $a^{(\tau)}, b^{(\tau)}$ с частично известными w, w'. Перебирая не более чем 2^{16} возможных значений пары $\left(w, w^{'}\right)$, мы можем определить ключ $k^{(\tau)}$, решая эту систему.

Аналогично в случае, когда $\varphi(\vartheta) = \vartheta \oplus \alpha$, $\alpha \in \mathbb{F}_{2^n}$, из первого уравнения системы (13) следуют равенства

$$f_k\,(w_i)\,a^{(\tau)} \oplus (f_k\,(w_i) \oplus \alpha)\,b^{(\tau)} = u_i^{(\tau)},$$
$$f_k\,(w_{i+1})\,a^{(\tau)} \oplus (f_k\,(w_{i+1}) \oplus \alpha)\,b^{(\tau)} = u_{i+1}^{(\tau)}.$$

Полагая

$$f_k\,(w_i) \oplus f_k\,(w_{i+1}) = w, \quad u = u_i^{(\tau)} \oplus u_{i+1}^{(\tau)},$$

получаем равенство

$$w\left(a^{(\tau)} \oplus b^{(\tau)}\right) = u$$

(с частично известным w). Выражая $a^{(\tau)}$ через $b^{(\tau)}$, из (13) получаем полиномиальное уравнение относительно $b^{(\tau)}$. Хотя такое уравнение может не иметь эффективного решения, мы получаем полезную информацию о производном ключе.

Использование рассеивающего преобразования Λ в формуле $c_i(k) = \Lambda(w * a) * b$ делает описанный метод неприменимым, поскольку в таком случае каждый байт вектора $\overline{\Lambda(w * a)}$ будет зависеть от каждого байта вектора $\overline{w * a}$ (как при максимально рассеивающем преобразовании в шифрсистеме «Кузнечик»). Отметим, что по той же причине для «разрушения» байтовой структуры рассеивающее преобразование применяется и в формуле (5). Использование неявной зависимости констант $c_i(k)$ и $c_j(k)$ иным образом не представляется возможным.

Предположим, что тем или иным образом стал известен ключ $k^{(\tau)}$. Насколько сложно тогда получить ключ k? Поскольку компоненты ключа $k^{(\tau)}$ вычисляются по формуле (5) как элементы $a^{(\tau)} = L_1(a, b)$, $b^{(\tau)} = L_2(a, b)$ заданных таблично ортогональных латинских квадратов L_1, L_2, найти ключ k можно лишь следующим образом. Будем перебирать возможные значения одной из переменных, например a, и находить соответствующие значения b из (6). Средняя сложность такого метода оценивается величиной $O(2^n)$. Учитывая сложность определения ключа $k^{(\tau)}$, можно сделать вывод, что предложенный метод выбора производного ключа не опасен.

$\mathbb{CS}$ является блочной криптосистемой. Применимы ли к ней известные методы анализа блочных шифров?

Линейный и разностный методы используются для анализа итерационных блочных шифров. Для их реализации требуется значительное число известных пар блоков открытых и шифрованных текстов, полученных в результате шифрования на общем ключе. Та же ситуация имеет место при атаках различения. Для анализа многораундовых шифров применяются также атаки со связанными ключами. Известные версии всех этих методов не работают в нашем случае, поскольку $\mathbb{CS}$ построена не по итерационному принципу и каждое сообщение шифруется на своём ключе, а связь между ключами выражается сложной неявной зависимостью.

В алгебраических атаках исследуется зависимость между битами блоков открытых и шифрованных текстов, полученная путём линеаризации нелинейных перемешивающих преобразований S-боксов. Задача нахождения ключа сводится к составлению и решению системы уравнений, число неизвестных в которой растёт экспоненциально от размеров S-боксов. В нашем случае потребуется, в частности, линеаризовать нелинейное перемешивающее преобразование $(\mathbb{F}_{2^8})^2 \to \mathbb{F}_{2^8}$, реализуемое квазигрупповой операцией $\overline{*}$ (аналог S-бокса). Число неизвестных в соответствующей системе уравнений будет непомерно велико.

Приведённый анализ позволяет сделать вывод о том, что предлагаемые варианты выбора параметров криптосистемы $\mathbb{CS}$ не имеют «видимых» слабостей в различных вариантах атак с выбранным открытым текстом. Более детальный анализ шифрсистемы возможен лишь при конкретном выборе параметров ортогональных латинских квадратов L_1, L_2, рассеивающего преобразования Λ и соотношения, связывающего константы $c_i(k)$ и $d_i(k)$.

Каковы возможности проведения противником активных атак? Как, наблюдая ряд аутентифицированных сообщений, построить подделку, т. е. новое сообщение, снабжённое корректной имитовставкой?

Для этого нужно хотя бы что-то знать о следующем производном ключе, например то, что он — тот же, что и предыдущий, иначе корректную имитовставку можно лишь угадывать. Но в силу того, что для $\mathbb{CS}$ выполняется условие (d), для противника следующий производный ключ выбирается случайно равновероятно из всего множества возможных ключей. Даже имея перехват многих сообщений, зашифрованных на общем основном ключе, противник не сможет ограничить для себя выбор следующего

производного ключа. Зная открытые и шифрованные тексты ряда сообщений, можно попытаться вычислить текущий производный ключ из системы уравнений (13). Но, как уже отмечено, эта задача сложна, даже если противник имеет возможность подбирать открытые тексты. Таким образом, успех в активной атаке возможен лишь в тех случаях, когда противник умеет определять производные ключи (ещё лучше, если основной ключ). Отсюда следует, что для проведения успешных активных атак противник должен уметь решать те же задачи, что и при проведении пассивных атак.

Рассмотрим теперь вопрос об оценке стойкости криптосистемы $\mathbb{CS}$ к атакам на основе шифртекста. Речь идёт о возможности решения противником следующих задач:

Задача 1: по шифртексту произвольного сообщения найти его открытый текст.

Задача 2: подменить перехваченное шифрованное сообщение таким образом, чтобы модифицированное сообщение было принято получателем как аутентичное.

Задача 3: составить такое поддельное шифрованное сообщение и имитировать его передачу от легального пользователя, чтобы оно было принято получателем как аутентичное.

В этих задачах криптосистема $\mathbb{CS}$ может рассматриваться как код аутентификации с секретностью при одноразовом использовании секретного ключа, поскольку противник не имеет практической возможности учесть взаимную зависимость производных ключей, обладая лишь перехватом шифртекстов. В этой постановке указанные задачи решались в [5, 6]. Результат решения задач 1–3 сформулирован в [6]. В [1] этот результат продублирован для варианта криптосистемы (теорема 2). Приведём здесь эти результаты, которые имеют место для каждого из рассмотренных в данной статье вариантов выбора параметров криптосистемы $\mathbb{CS}$, дополнив их ещё одной оценкой.

Обозначим через p_{im} и p_{sub} вероятности успеха имитации и подмены (т. е. вероятности успешного решения задач 3 и 2 соответственно), а $p_{\mathrm{im}}^{(N)}$ — вероятность успеха имитации при условии, что противник наблюдает сообщения $m^{(1)}, \ldots, m^{(N)}$, зашифрованные с помощью общего основного ключа k.

Теорема 1. Пусть ключ k криптосистемы $\mathbb{CS}$ выбирается случайно равновероятно, а распределение вероятностей $\mathcal{P}_X$ на множестве открытых текстов $\mathcal{X}$ таково, что для действительных чисел α, β, σ, удовлетворяющих неравенствам $0 < \alpha$, $\beta < 1 \leqslant \sigma$, выполняются условия $\alpha \leqslant p_X(x) \leqslant \beta$, $\beta/\alpha \leqslant \sigma$ для каждого $x \in \mathcal{X}$. Тогда $\mathbb{CS}$ реализует ε-совершенное шифрование очередного текста $s^{(\tau)}$, $\tau \geqslant 1$, состоящего из $r^{(\tau)} > 1$ блоков, для $\varepsilon < \sigma 2^{-n}$. При этом $p_{\mathrm{im}} = 2^{-n}$, $p_{\mathrm{sub}} \leqslant r^{(\tau)}\sigma 2^{-n}$. Кроме того, при $N < 2^n$ справедлива оценка $p_{\mathrm{im}}^{(N)} \leqslant 2^n / (2^{2n} - N)$.

Приведённая в теореме 1 оценка величины $p_{\mathrm{im}}^{(N)}$ дополняет утверждение теоремы 2 в [1]. Она следует из того, что рассматриваемые варианты выработки производного ключа гарантируют свойство (c).

Теорема 1 свидетельствует о доказуемой стойкости криптосистемы $\mathbb{CS}$ к атакам на основе шифртекста, так как при подходящих значениях n, r, σ гарантируется как ε-совершенное шифрование, так и достаточно малые значения вероятностей успеха активных атак. Так, при $n = 128$, $\sigma = 2^{32}$, $r \leqslant 2^{32}$ криптосистема реализует ε-совершенное шифрование текстов длины не больше 2^{32} блоков для $\varepsilon < 2^{-96}$ и их имитозащиту со значениями $p_{\mathrm{im}} = 2^{-128}$, $p_{\mathrm{sub}} < 2^{-64}$. В частности, $p_{\mathrm{im}}^{(N)} < 1/(2^{128} - 1)$ при $N = 2^{128} - 1$.

Получение этих оценок для криптосистем, реализующих шифрование информации и её аутентификацию, оказалось возможным вследствие того, что для шифрования используется не традиционная блочная шифрсистема, а параметризованная ключом

система отображений, порождённая комбинаторной структурой, в данном случае — кодом Рида — Соломона.

В заключение отметим, что предложенные варианты параметров модели криптосистемы выбирались, исходя из соображений разумной минимизации их сложности, не приводящей к «видимым» криптографическим недостаткам. Можно предложить варианты усиления схемы, усложняющие задачу нахождения ключа в атаке с выбранным открытым текстом, для которых остаются в силе оценки сформулированной теоремы.

ЛИТЕРАТУРА

1. *Зубов А. Ю.* Криптосистема шифрования с аутентификацией на основе кода аутентификации с секретностью // Прикладная дискретная математика. 2019. № 43. С. 60–69.
2. *Сачков В. Н.* Введение в комбинаторные методы дискретной математики. М.: МЦНМО, 2004. 424 с.
3. *McGrew D. and Viega J.* The security and performance of Galois/Counter mode of operation // LNCS. 2004. V. 3348. P. 343–355.
4. *Viega J. and McGrew D.* Galois/Counter mode (GCM) overview fibre channel security protocols. Cisco Systems Inc., 2005. 32 p.
5. *Зубов А. Ю.* Почти совершенные шифры и коды аутентификации // Прикладная дискретная математика. 2011. № 4 (14). С. 28–33.
6. *Зубов А. Ю.* О понятии ε-совершенного шифра // Прикладная дискретная математика. 2016. № 3 (33). С. 45–52.

REFERENCES

1. *Zubov A. Yu.* Kriptosistema shifrovaniya s autentifikatsiey na osnove koda autentifikatsii s sekretnost'yu [Authentication encryption based on authentication code with secrecy]. Prikladnaya Diskretnaya Matematika, 2019, no. 43, pp. 60–69. (in Russian)
2. *Sachkov V. N.* Vvedenie v kombinatornye metody diskretnoy matematiki [Introduction to Combinatorial Methods of Discrete Mathematics]. Moscow, MCCME Publ., 2004. 424 p. (in Russian)
3. *McGrew D. and Viega J.* The security and performance of Galois/Counter mode of operation. LNCS, 2004, vol. 3348, pp. 343–355.
4. *Viega J. and McGrew D.* Galois/Counter mode (GCM) overview fibre channel security protocols. Cisco Systems Inc., 2005. 32 p.
5. *Zubov A. Yu.* Pochti sovershennye shifry i kody autentifikatsii [Almost perfect ciphers and authentication codes]. Prikladnaya Diskretnaya Matematika, 2011, no. 4 (14), pp. 28–33. (in Russian)
6. *Zubov A. Yu.* O ponyatii ε-sovershennogo shifra [On the concept of a ε-perfect cipher]. Prikladnaya Diskretnaya Matematika, 2016, no. 3 (33), pp. 45–52. (in Russian)