**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