

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  $\psi(g^t): V_{2^{13}} \rightarrow V_{128}$  со свойством полного перемешивания входных данных. При  $6 \leq t \leq 100$  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*-боксов. 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  $2^n$ ;  $\bar{X}$  — 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.

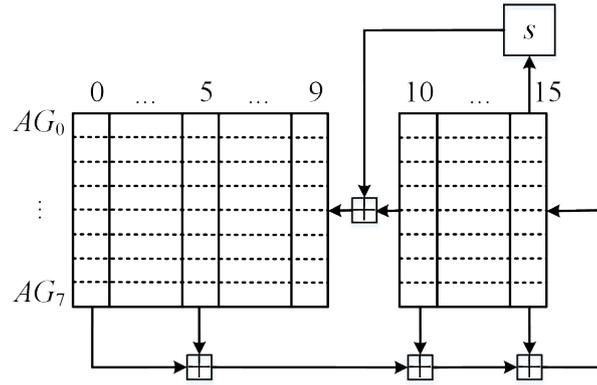
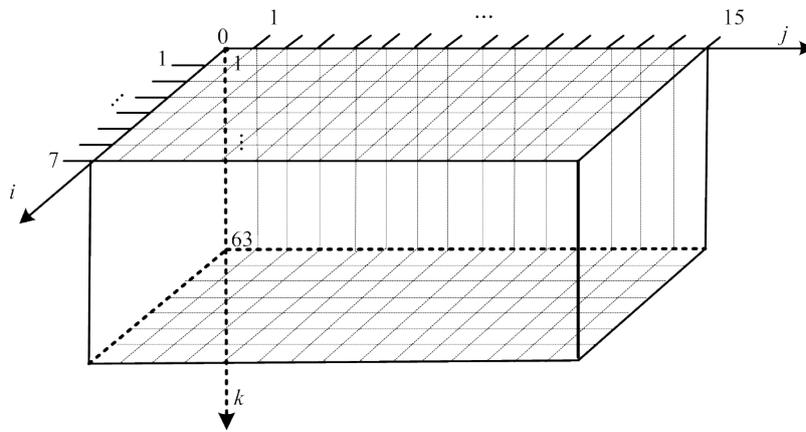
Булева функция называется вполне перемешивающей [2], если она существенно зависит от каждой переменной. Отображение  $V_n \rightarrow V_m$  называется вполне перемешивающим, если каждая его координатная функция вполне перемешивающая.

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

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

Построим алгоритм генерации  $r$ -битового ККЦ для информационного  $l$ -битового блока, где  $l = 2^{13}$  бит (1 кбайт),  $r = 128$ . Алгоритм реализует функцию  $\psi(g^t): V_{2^{13}} \rightarrow V_{128}$ , где  $g: V_{2^{13}} \rightarrow V_{2^{13}}$  — преобразование регистрового типа множества состояний схемы из восьми идентичных аддитивных генераторов  $AG_0, \dots, AG_7$ , модифицированное с помощью преобразования  $s(a_0, \dots, a_7)$  (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):  $0 \leq i < 8, 0 \leq j < 16, 0 \leq k < 64$ .

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  $t \geq 0$  обозначим  $V_{8,16,64}^{(t)} = \{x_{i,j,k}^{(t)}\}$ , или матрицей  $M_A^{(t)} = (X_{i,j}^{(t)})$  над  $\mathbb{Z}_{2^{64}}$ , где  $\bar{X}_{i,j}^{(t)} = (x_{i,j,0}^{(t)}, \dots, x_{i,j,63}^{(t)})$  — состояние на  $t$ -м такте  $j$ -й ячейки  $AG_i$ .

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

$$S^{(t)} = (s_0^{(t)}(\omega^{(t)}) \dots s_7^{(t)}(\omega^{(t)})), \quad (1)$$

где  $s_0^{(t)}(\omega^{(t)}) = s(\omega^{(t)})$ ,  $s_j^{(t)}(\omega^{(t)}) = s(s_{j-1}^{(t)}(\omega^{(t)}) \oplus \omega^{(t)})$ ,  $j = 1, \dots, 7$ .

Функция переходов автомата задана равенствами

$$(X_{i,0}^{(t+1)}, \dots, X_{i,15}^{(t+1)}) = (Y_{i,1}^{(t)}, \dots, Y_{i,15}^{(t)}, f(Y_{i,0}^{(t)}, \dots, Y_{i,15}^{(t)})), \quad 0 \leq i < 8,$$

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

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

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

## 2. Исследование множества образов отображения $z(s): V_8 \rightarrow V_{64}$

Вектор  $S^{(t)}$  используется в алгоритме как псевдослучайный сдвиг векторов  $X_{i,10}^{(t)}$ ,  $0 \leq i < 8$ . 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  $S(0, \dots, 0) = (0, \dots, 0)$  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  $S(y) = (y_0, y_1, \dots, y_7)$ , где  $y_0 = s(y)$ ,  $y_j = s(y_{j-1} \oplus y)$ ,  $j = 1, \dots, 7$ , и посчитано число различных байтов, составляющих  $S(y)$ . В табл. 1 приведено число  $\nu_r$  векторов  $S(y)$  (для 256 возможных значений  $y$ ), состоящих ровно из  $r$  различных байтов среди  $y_0, \dots, y_7$ ,  $r = 1, \dots, 8$ ; результаты получены для  $s$ -боксов размера  $8 \times 8$  известных блочных шифров.

Т а б л и ц а 1

Число  $\nu_r$  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 $\pi_0$	1	1	1	2	2	0	0	249
Kalyna $\pi_1$	1	1	0	0	2	2	0	250
Kalyna $\pi_2$	1	1	0	3	1	0	1	249
Kalyna $\pi_3$	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  $2^{t-2r}$ . 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  $t \geq 6$  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  $6 \leq t \leq 100$ .

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  $s$ -блока. 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.

Т а б л и ц а 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  $s$ -блоков 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](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.