Ежегодная международная научно-практическая конференция
«РусКрипто’2022»

Об одном классе алгоритмов контроля целостности больших блоков данных

Дмитрий Бобровский,
Владимир Фомичёв,
Дмитрий Задорожный,
Алиса Коренева,
Алексей Курочкин

КОД безопасности

Актуальность

Проблема снижения ресурсоемкости вычислений для контроля целостности:

- Динамический контроль больших объемов данных.
- Контроль целостности (КЦ) исполняющей среды функционирования.
- Оперативный аудит целевых систем.

Используемые подходы для КЦ:

Вычислительно сложные

- хэш-функции (SHA, MD, ГОСТ34.11-2018);
- хэш-функции с ключом (HMAC);
- блочные шифры в режиме выработки имитовставки (CMAC).

Высокопроизводительные

- некриптографические методы с использованием кодов, обнаруживающих и/или исправляющих ошибки (коды Хэмминга, циклические коды (CRC [1]) и др.).

Актуальная задача – построение альтернативных алгоритмов КЦ, компромиссных в части криптографических характеристик и скорости вычисления.

[1] Stigge, Martin, H. Plötz, W. Müller and Jens-Peter Redlich. “Reversing CRC – Theory and Practice.” (2006).

Научный фундамент

1. Фомичёв В.М., Авезова Я.Э., Коренева А.М., Кяжин С.Н. Примитивность и локальная примитивность орграфов и неотрицательных матриц. Дискретный анализ и исследование операций, 2018, 25(3), с.95-125.
2. Коренева А.М. Оценки экспонентов перемешивающих орграфов регистровых преобразований, используемых в системах защиты информации: дис. канд. физ.-мат. наук. МГУ имени М.В. Ломоносова, Москва, 2018. Научный руководитель – д.ф.-м.н., профессор Фомичёв В. М.
3. Фомичев В. М., Коренева А. М., Набиев Т. Р. О новом алгоритме контроля целостности данных, РусКрипто'20, 2020.
4. Фомичев В. М., Коренева А. М., Набиев Т. Р. Характеристики алгоритма контроля целостности данных на основе аддитивных генераторов и s-боксов, ПДМ. Приложение, 2020, № 13, с.62–66.
5. Бобровский Д.А., Задорожный Д.И., Коренева А.М., Набиев Т.Р., Фомичев В.М. О контроле целостности хранимых данных с использованием хэширования, РусКрипто'21, 2021.
6. Бобровский Д.А., Задорожный Д.И., Коренева А.М., Набиев Т.Р., Фомичев В.М. Экспериментальное исследование характеристик одного способа контроля целостности при хранении данных большого объёма, ПДМ. Приложение, 2021, № 14, с.71–74.
7. Fomichev V., Bobrovskiy D., Koreneva A., Nabiev T., Zadorozhny D. Data integrity algorithm based on additive generators and hash function. Springer. J Comput Virol Hack Tech (2021). URL: https://rdcu.be/cA7QI

Цель исследования

- Разработка класса высокопроизводительных алгоритмов КЦ для блока данных размера 1 КиБ и более.

Задачи

- Экспериментально проверить свойства класса алгоритмов.
- При выбранных параметрах оценить производительность и криптографические свойства алгоритма из класса.
- Сравнить характеристики алгоритма из класса с другими алгоритмами.

Алгоритм *AG-S*

- Алгоритм генерации $2r$-битового кода КЦ для блока длины l бит.
- Генератор AG-S состоит из s-боксов и h идентичных аддитивных генераторов (регистров сдвига длины $n \in \mathbb{N}$) AG_0-S, …, AG_{h-1}-S.
- В каждой ячейке любого регистра записано число из Z_{2^r} (или, что равносильно, вектор из V_r).
- Множество состояний генератора представимо матрицей $M^{(t)} = \left(X_{i,j}^{(t)}\right)$ над кольцом Z_{2^r}, где $X_{i,j}^{(t)}$ – число, записанное в t-м такте в j-й ячейке AG_i-S; двоичная запись этого числа есть
  - $\bar{X}_{i,j}^{(t)} = \left(x_{(i,j,0)}^{(t)}, \dots, x_{(i,j,r-1)}^{(t)}\right)$,
- $0 \le i < h$,
- $0 \le j < n$.

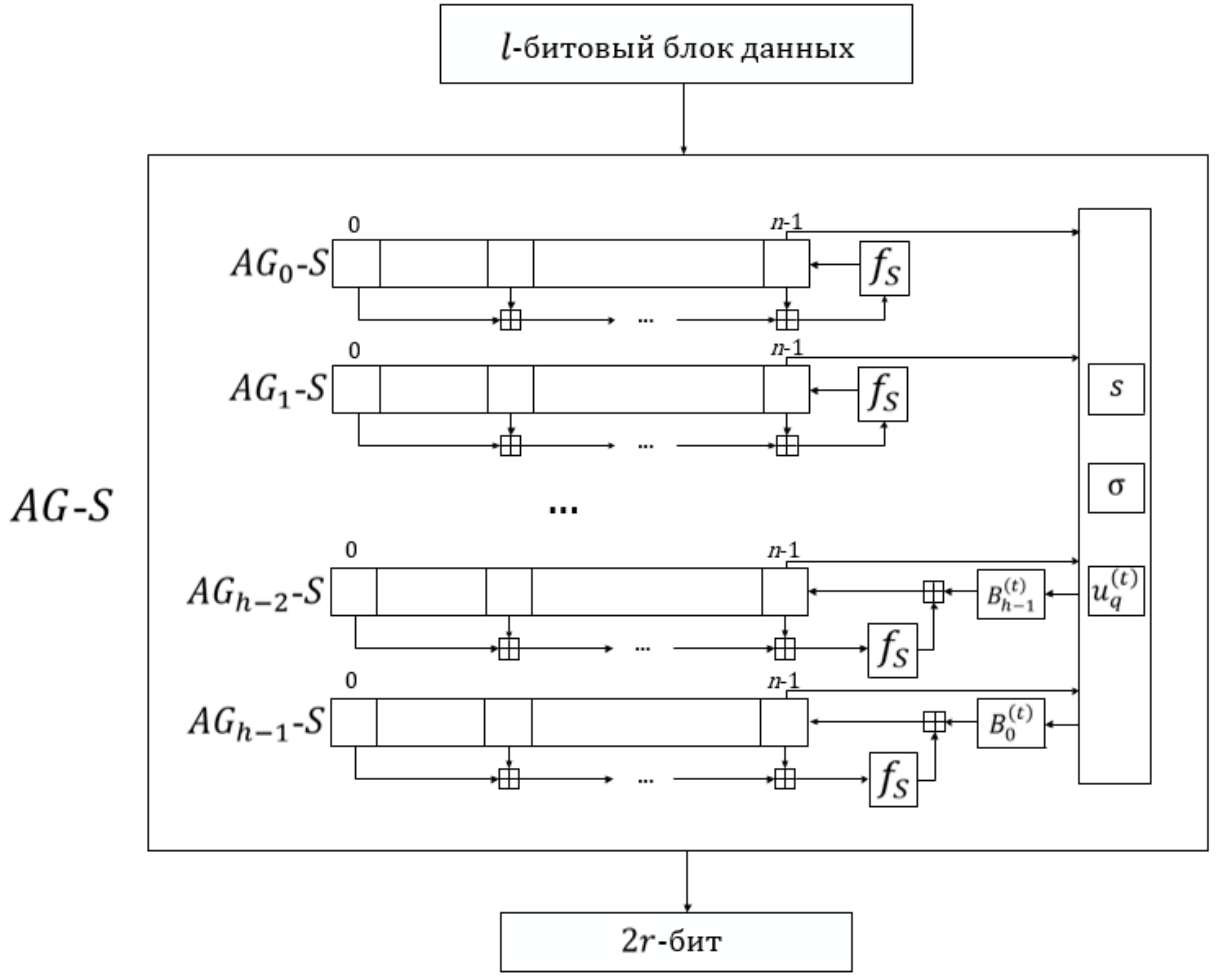


Рис.1. Блок-схема алгоритма *AG-S*

Преобразование регистра сдвига над кольцом Z_{2^r}

Алгоритм генерации AG-S вычисляет функцию $\psi\left(g^t_{S_{r,q}}\left(M^{(0)}\right)\right)$, где $\psi : V_l \to V_m$, t – число итераций, $g_{S_{r,q}}: V_r \to V_r$ – базовое преобразование множества состояний генератора AG-S, зависящее от преобразования $S_{r,q}: V_r \to V_r$ – преобразование с помощью $\frac{r}{q}$ идентичных s-боксов:

$$S_{r,q}(x_0, \dots, x_{r-1}) = s\left(x_{r-1}, \dots, x_{r-q}\right)||s\left(x_{r-q-1}, \dots, x_{r-2q}\right)|| \dots ||s\left(x_{q-1}, \dots, x_0\right),$$

$s\left(a_0, \dots, a_{q-1}\right)$ – функция s-бокса размера q×q бит.

Каждый AG_i-S, $0 \le i < h$, реализует регистровое преобразование φ_S длины n с нелинейной обратной связью $f_S(X_0, \dots, X_{n-1})$, зависящей от функции $S_{r,q}$:

$$\varphi_S(X_0, \dots, X_{n-1}) = \left(X_1, \dots, X_{n-1}, f_S(X_0, \dots, X_{n-1})\right).$$
$$f_S(X_0, \dots, X_{n-1}) = \mathrm{Int}_r\left(S_{r,q}\left(\mathrm{Vec}_r(c_0X_0 \boxplus \cdots \boxplus c_{n-1}X_{n-1})\right)\right),$$

где $c_0, \dots, c_{n\_1} \in Z_{2^r}$, $c_0 = c_{n\_1} = 1$,

Vec_r — биекция $Z_{2^r} \leftrightarrow V_r$, определяющая двоичное r-разрядное представление числа $X \in Z_{2^r}$,

Int_r — обратная к Vec_r функция,

$\boxplus$ — операция сложения по модулю 2^r.

Преобразование состояния генератора

В t-м такте выполняется преобразование $g_S\left(M^{(t)}\right) = M^{(t+1)}$.

Для этого выполняются регистровые преобразования φ_S каждой строки матрицы $M^{(t)}$, затем состояние $(n-1)$-й ячейки i–го регистра суммируется с определенным ниже числом $B_i^{(t)}$, $i = 0, \dots, h-1$:

$$g_S\left(M^{(t)}\right) = M^{(t+1)} = \begin{pmatrix} X_{0,1}^{(t)} & \dots & X_{0,n-1}^{(t)} & B_1^{(t)} \boxplus f_S(X_{0,0}^{(t)}, \dots, X_{0,n-1}^{(t)}) \\ \dots & \dots & \dots & \dots \\ X_{h-2,1}^{(t)} & \dots & X_{h-2,n-1}^{(t)} & B_{h-1}^{(t)} \boxplus f_S(X_{h-2,0}^{(t)}, \dots, X_{h-2,n-1}^{(t)}) \\ X_{h-1,1}^{(t)} & \dots & X_{h-1,n-1}^{(t)} & B_0^{(t)} \boxplus f_S(X_{h-1,0}^{(t)}, \dots, X_{h-1,n-1}^{(t)}) \end{pmatrix},$$

где числа $B_0^{(t)}, \dots, B_{h-1}^{(t)}$ вычисляются с помощью чисел $X_{0,n-1}^{(t)}, \dots, X_{h-1,n-1}^{(t)}$, s-бокса и нескольких случайных чисел, сгенерированных заранее.

Формирование кода КЦ

Код контроля целостности (ККЦ) есть пара чисел из Z_{2^r}:

$$\psi\left(g_S^t(M^{(0)})\right) = \left(\left(\sum_{i=0}^{\frac{h}{2}-1} \varepsilon_{2i} X_{2i,n-1}^{(t)}\right) \bmod 2^r, \left(\sum_{i=0}^{\frac{h}{2}-1} \varepsilon_{2i+1} X_{2i+1,n-1}^{(t)}\right) \bmod 2^r\right),$$

где $\varepsilon_0 = \varepsilon_{h-1} = 1, \varepsilon_i = (2^i+1) \bmod 2^r, i = 1,\ldots,h-2.$

Характеристики алгоритма из класса (1)

Оценка характеристик алгоритма генерации ККЦ выполнена для следующего наборов параметров:

$h = 8, n = 16, r = 64$, размер s-бокса 8×8,

векторы $\bar{B}_1^{(t)}, \dots, \bar{B}_6^{(t)}$ нулевые,

$$\bar{B}_7^{(t)} = \alpha_7 || \left(s\left(\sigma\left(\bar{X}_{7,15}^{(t)}\right), \dots, \sigma\left(\bar{X}_{0,15}^{(t)}\right)\right) \oplus u_8^{(t)} \right) || \left(s\left(\sigma\left(\bar{X}_{7,15}^{(t)}\right), \dots, \sigma\left(\bar{X}_{0,15}^{(t)}\right)\right) \oplus u_8^{(t)} \right),$$

$$\bar{B}_0^{(t)} = \alpha_0 || \left(s\left(\sigma\left(\bar{X}_{0,15}^{(t)}\right), \dots, \sigma\left(\bar{X}_{7,15}^{(t)}\right)\right) \oplus u_8^{(t)} \right) || \left(s\left(\sigma\left(\bar{X}_{0,15}^{(t)}\right), \dots, \sigma\left(\bar{X}_{7,15}^{(t)}\right)\right) \oplus u_8^{(t)} \right),$$

где $\alpha_0 = 0\text{x}7\text{B}4\text{DCFE}34\text{FDE}$, $\alpha_7 = 0\text{x}5\text{B}4\text{AC}901\text{BCDF}$; $\sigma(x_0, \dots, x_{63}) = x_0 \oplus \dots \oplus x_{63}$ — булева функция четности веса двоичной строки $(x_0, \dots, x_{63})$;

$u_8^{(t)}$=($u^{(t)}$,…,$u^{(t)}$) есть битовая строка длины 8, равная (0,…,0) или (1,…,1) при фиксированном t,

$u^{(t)} = \sigma(\bar{X}_{1,15}^{(t)}) \oplus \dots \oplus \sigma(\bar{X}_{7,15}^{(t)})$ — булева функция четности веса состояния (исключая компоненту $\bar{X}_{0,15}^{(t)}$) в такт t.

Характеристики алгоритма из класса (2)

$$\varphi_S(X_0, \dots, X_{15}) = \left(X_1, \dots, X_{15}, \mathrm{Int}_{64}\left(S_{64,8}\left(\mathrm{Vec}_{64}(\Sigma(X_0, \dots, X_{15}))\right)\right)\right),$$

$$\Sigma(X_0, \dots, X_{15}) = X_0 \boxplus X_2 \boxplus \left((X_4 \gg 9) \oplus (X_4 \ll 27)\right) \boxplus X_6 \boxplus X_8 \boxplus \left((X_{10} \gg 25) \oplus\right.$$

Результаты экспериментальных исследований (1)

Измерена производительность (CpB, такты/байт) генерации кодов КЦ разными алгоритмами при входных блоках 1 КиБ (реализации из OpenSSL, Crypto++). Выбранный алгоритм из класса производительнее от 2 до 73 раз, чем другие известные алгоритмы расчета кода КЦ.

Таблица 1. Характеристики производительности

№	Алгоритм	CpB	№	Алгоритм	CpB	№	Алгоритм	CpB	№	Алгоритм	CpB
1	ag-s	3,25	9	sha3-256	8,052	17	sha512	11,317	25	streebog256	18,645
2	crc16	6,472	10	sha1	8,923	18	ripemd	11,463	26	streebog512	19,661
3	crc32	6,67	11	sha3-384	9,141	19	sha512-224	11,753	27	md_gost12_512	20,313
4	crc64	6,829	12	ripemd160	9,68	20	sha224	11,971	28	hmac-sha1	20,48
5	md5	7,69	13	sha3-512	10,075	21	sha384	12,115	29	md_gost94	36,201
6	blake2s256	7,763	14	blake2b512	10,447	22	md5-sha1	12,913	30	cmac-gost	88,282
7	shake128	7,835	15	rmd160	10,737	23	sha512-256	13,711	31	md_gost12_256	162,071
8	shake256	7,98	16	sm3	10,882	24	sha256	16,013	32	sha3-224	239,624

ПЭВМ Intel Core i5-8600 3.1 GHz.

Результаты экспериментальных исследований (2)

- С помощью матрично-графового подхода установлены положительные свойства перемешивания, присущие классу алгоритмов AG-S в целом.
- Экспериментально установлено, что ряд характеристик функции $\psi\left(g_S^7(M^{(0)})\right)$ близок к характеристикам случайных функций.
- Проведено исследование алгоритмов с помощью набора тестов *SMHasher*[1]. Все тесты пройдены.

Есть одно но! При выбранных параметрах имеется несложная атака по построению второго прообраза.

Объяснимо свойствами выбранной функции модификации АГ и представляется вполне поправимым.

[1] Набор тестов SMHasher: https://github.com/rurban/smhasher

Выводы

- Предложен класс *AG*-S высокопроизводительных алгоритмов генерации кода КЦ на основе аддитивных генераторов и S-боксов.
- Установлены положительные свойства (скоростные характеристики, перемешивание), присущие классу алгоритмов.
- Экспериментально исследована производительность алгоритма *AG*-S при выбранных параметрах. На входном материале 1КиБ алгоритм *AG-S* превосходит по производительности от 2 до 73 раз известные алгоритмы.

Перспективные направления исследования:

- Определение области значений параметров предложенного класса алгоритмов КЦ, при которых построение коллизий является вычислительно сложной задачей.

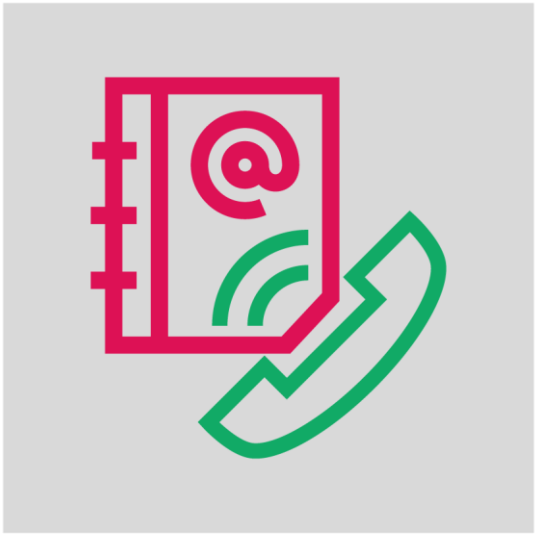
Спасибо за внимание!

Вопросы

Контактная информация

Дмитрий Бобровский

d.bobrovskiy@securitycode.ru



Построение атаки нахождения второго прообраза при выбранных параметрах

Пусть $M^{(0)}$ – блок данных, ККЦ которого есть $Q = \psi\left(M^{(7)}\right) = \psi\left(g_S^7\left(M^{(0)}\right)\right)$, при этом $Q = \psi\left(g_S^1\left(M^{(6)}\right)\right) = \psi\left(g_S^2\left(M^{(5)}\right)\right) = \ldots = \psi\left(g_S^6\left(M^{(1)}\right)\right)$.

Для получения из $M^{(k)}$ значения $M^{(k-1)}$ необходимо не более $2^8 \times 8 \times 8 = 2^{14}$ операций вычисления прообразов байтовой подстановки, а значит для вычисления прообраза $M^{(0)}$ из $M^{(7)}$ необходимо не более 7×2^{14} операций вычисления прообразов байтовой подстановки.

Образ Q зависит только от значений $X_{i,15}^{(7)}$, $0 \le i < 8$, т.е. можно построить такой блок $M'^{(7)} = \left(X'^{(7)}_{i,j}\right)$, для которого верно $\psi\left(M^{(7)}\right) = \psi\left(M'^{(7)}\right)$, например, взяв блок $M'^{(7)}$, для которого $X_{i,15}^{(7)} = X'^{(7)}_{i,15}$, а любая другая ячейка блока $M'^{(7)}$ есть случайное значение из V_{64}. Выбрав блок $M'^{(7)}$, найдем $M'^{(0)}$ такое, что $\psi\left(g_S^7\left(M^{(0)}\right)\right) = \psi\left(g_S^7\left(M'^{(0)}\right)\right)$, это требует не более 7×2^{14} операций.