infotecs

Сравнение вычислительной сложности алгоритмов выработки производных ключей

Бородин Михаил
ОАО Инфотекс

РусКрипто, 2017

Исследуемые механизмы

- OMAC
- HMAC
- ACPKM-CTR-Kuznyechik
- KDF_GOSTR3411_2012_256
- KDF_TREE_GOSTR3411_2012_256
- Solo_Hash

Кузнечик и Стрибог

Алгоритм зашифрования:

$E_K(a) = X[K_{10}]LSX[K_9]\ldots LSX[K_2]LSX[K_1](a),$

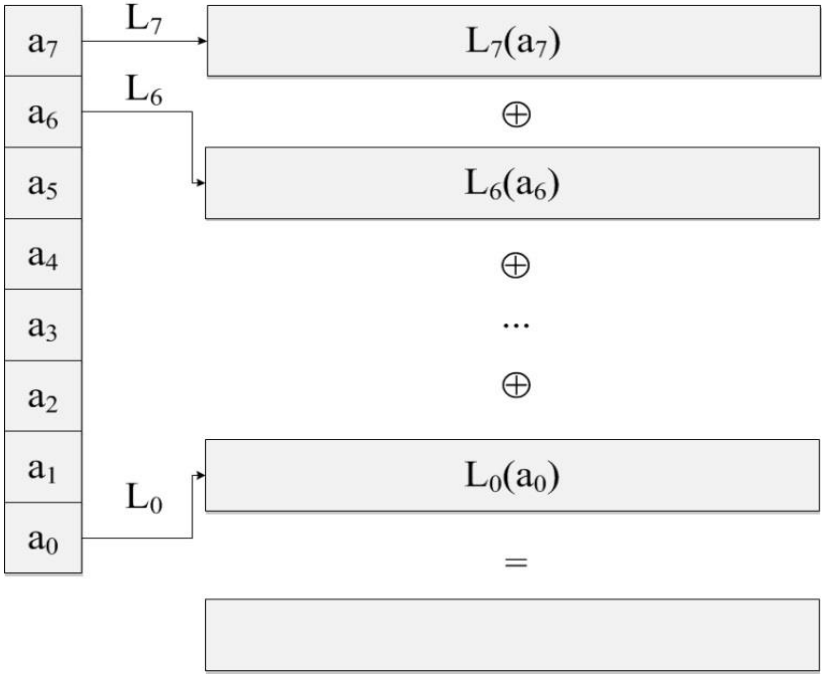
Функция сжатия: $V_{512} \times V_{512} \rightarrow V_{512}$

$g_N(h,m) = E(LPS(h \oplus N),m) \oplus h \oplus m,$

$E(K,m) = X[K_{13}]LPSX[K_{12}]\ldots LPSX[K_2]LPSX[K_1](m).$

$K_1 = K; \quad Ki = LPSX(Ki_{-1} \oplus C_{i-1)}, \quad i = 2, \ldots, 13.$

Сложность преобразования LS



Сложность:

- Операции выбора 64-х битного вектора по адресу из таблицы;
- Побитовое сложение 64-х битных векторов (XOR);

Остальными операциями пренебрегаем

	LS «Стрибог»(512 бит)	LS «Кузнечик» (128бит)
Позиционирование	64	32
XOR	56	30
Всего	120	62

Корректность модели

	«Стрибог» сжатие (512 бит)	«Кузнечик» (128бит)	«Кузнечик» 4 блока (512бит)	Ключевая развертка
Позиционирование	1600	288	1152	1024
XOR	1400	270	1080	960
Всего	3000	558	2232	1984
Скорость	92 МБ/сек	125 МБ/сек		---

$$\frac{2232}{3000} = \frac{93}{125} \approx \frac{92}{125}$$

HMAC и OMAC

K – Ключ;
$X = X_1||X_2|| \dots ||Xq$ – данные, $X_i \in Vn$.

HMAC:

$$HMAC_K^n(X) = Hash^n((K \oplus opad)||Hash^n((K \oplus ipad)||X)),$$

$opad, ipad \in V_{512}$, $Hash^n$ – «Стрибог», $n = 256$ или $n = 512$.

OMAC:

$$C_0 = 0^n; \quad C_i = E_K(X_i \oplus C_{i\_1}), \quad i = 1,2, \dots, q-1.$$
$$OMAC_K^n(X) = T_s(E_K(X_q \oplus C_{q\_1} \oplus K^*)),$$

при вычислении K^* используется зашифрование $E_K(0^n)$.
Для «Кузнечика» $n = 128$.

HMAC vs OMAC

Алгоритм

$$C_0 = 0;$$
$$z_0 = IV;$$
$$C_i = C_{i-1} + 1;$$
$$z_i = MAC^n_{MK}(z_{i-1}||C_i||P);$$
$$K = (z_1||z_2|| \dots ||z_{\lceil L/n \rceil}).$$

MK – мастер ключ;
n – размер выхода функции MAC;
IV– вектор инициализации;
L – битовая длина производного ключа;
P –сопутствующие данные;
$MAC^n_{mk}(X)$ – одна из двух функций $HMAC^n_{MK}(X)$ или $OMAC^n_{MK}(X)$.

	HMAC		OMAC без развертки		OMAC с развертки	
	Функций сжатия	Общее	Блоков «Кузнечика»	Общее	Блоков «Кузнечика»	Общее
Ключ 256 бит	8	24000	6 (+1)	3350 (+558)	9,5 + развертка(+ $E^n(0)$)	5300 (+558)
Ключ 512 бит	9	27000	12 (+1)	6700 (+558)	15,5 + развертка ($E^n(0)$)	8650 (+558)

АЛГОРИТМЫ

ACPKM-CTR-Kuznyechik

$K_0 = K;$
$K_{i+1} = E_{K^i}(D5) \,||\, E_{K^i}(D6).$

$i \in \{0,1, \dots, l-2\};$
$D5, D6 \in const\ V_{128}.$

KDF_GOSTR3411_2012_256

$KDF_{MK}^{256}(l,s) = HMAC_{MK}^{256}(0x01 \,||\, l \,||\, 0x00 \,||\, s \,||\, 0x01 \,||\, 0x00).$
l, s – сопутствующая информация

KDF_TREE_GOSTR3411_2012_256

$KDFTREE_{MK}^{256}(l,s,R) = \text{K}(1) \,||\, \text{K}(2) \,||\, \text{K}(3) \,||\, \dots;$
$K(i) = HMAC_{MK}^{256}([i]_b \,||\, l \,||\, 0x00 \,||\, s \,||\, [L]_b).$

l, s – сопутствующая информация; i – счетчик числа итераций;
$[i]_b$ – байтовое представление счетчика числа итераций

Solo_Hash

$Solo\_Hash^n(MK, IV, X) = Hash^n(MK||IV||X);$

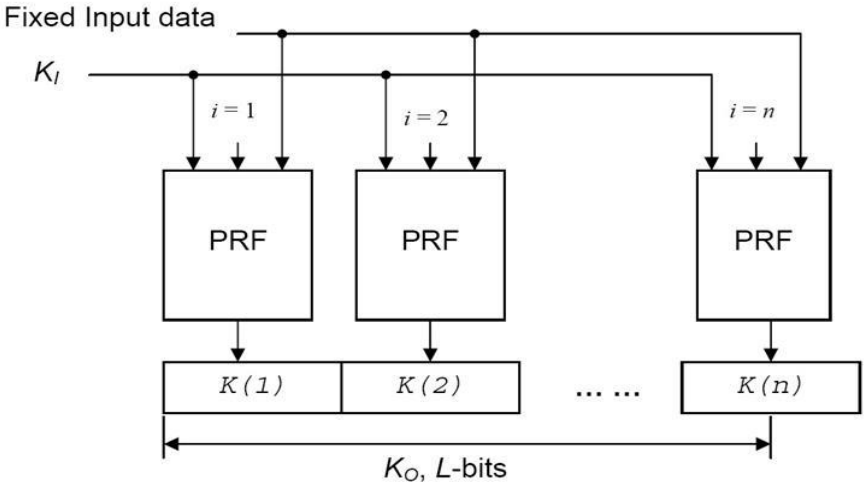
IV– вектор инициализации;
X– сопутствующая информация.

Итоговая таблица

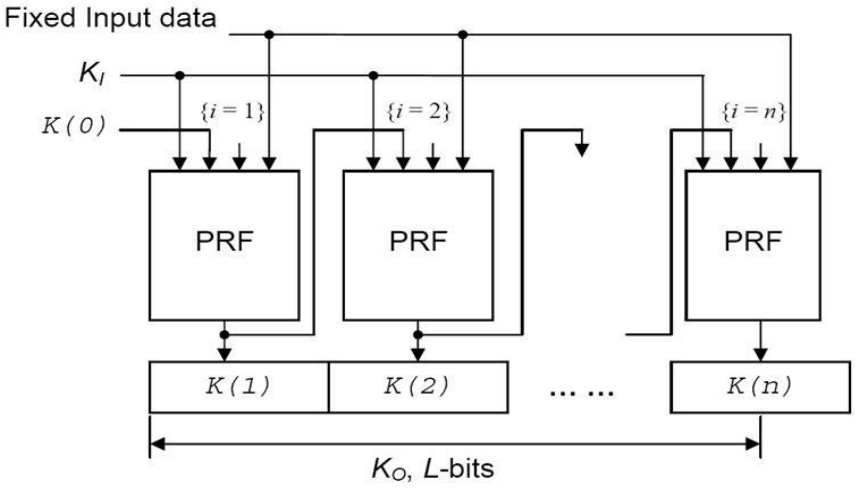
	256 бит			512 бит		
	F сжатия	Блоков «Кузнечика»	Общих	F сжатия	Блоков «Кузнечика»	Общих
Ключевая развертка	0	~3,5	1984	-	-	-
OMAC	0	6 (+1)	3350 (+558)	0	12 (+1)	6700 (+558)
OMAC + развертка	0	9,5 (+1)	5300 (+558)	0	~15,5 (+1)	8650 (+558)
HMAC	8	0	24000	9-10	0	27000
ACPKM-CTR-Kuznyechik +развертка	0	5,5	3100	0	~11	6200
KDF_GOSTR3411_2012_256	8	0	24000	16	0	48000
KDF_TREE_GOSTR3411_2012_256	8	0	24000	16	0	48000
Solo_Hash	3-4	0	12000	4	0	12000

NIST SP 800-108

CTR-схема



CBC-схема



	CTR-схема	CBC-схема
OMAC	—	1
HMAC	2	1

1. И.В. Лавриков, В.И. Рудской «О возможных подходах к построению механизмов выработки производных ключей и механизмов выработки псевдослучайных последовательностей», РусКрипто 2016;
2. Рекомендация ТК26 «Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования».

Спасибо за внимание!