Исследование характеристик одного корреляционного метода анализа

Фомин Андрей Викторович
Фомин Денис Бонославович

Национальный исследовательский университет «Высшая школа экономики»

Ученые подсчитали, что шансы реального существования столь откровенно абсурдного мира равняются одному на миллион. Однако волшебники подсчитали, что шанс «один на миллион» выпадает в девяти случаях из десяти.

Терри Пратчетт
Мор, ученик Смерти

При криптографических исследованиях поточных шифров часто используется теоретико-кодовый подход, связанный со сведением задачи восстановления ключа к задаче декодирования линейного кода:

При этом характеристики метода анализа зависят от используемого алгоритма декодирования.

LILI-128 — поточный шифр, представленный в рамках проекта NESSIE в 2000 году

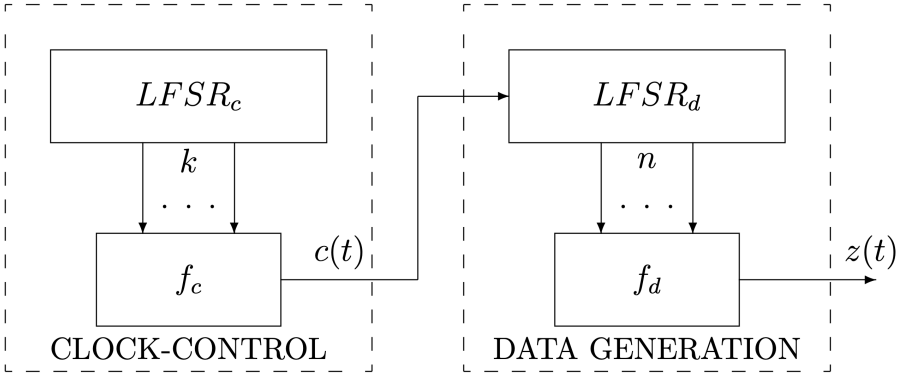

Рис.: Схематичное устройство шифра LILI-128

РСЛОС$_c$ реализует линейную рекуррентную последовательность над полем $\mathbb{F}_2$ с характеристическим многочленом

$$x^{39} + x^{35} + x^{33} + x^{31} + x^{17} + x^{15} + x^{14} + x^2 + x + 1.$$

В каждый такт работы t значения на позициях 12 и 20 подаются на вход функции f_c, значение которой определяет последовательность c_t по правилу:

$$c_t = f_c(x_{12}, x_{20}) = 2(x_{12}) + x_{20} + 1,$$

где сложение происходит в $\mathbb{Z}$. Откуда получаем, что $c_t \in \{1, 2, 3, 4\}$.

РСЛОС$_d$ реализует линейную рекуррентную последовательность над полем $\mathbb{F}_2$ с характеристическим многочленом

$$x^{89} + x^{83} + x^{80} + x^{55} + x^{53} + x^{42} + x^{39} + x + 1.$$

Последовательность c_t, вырабатываемая управляющим генератором, показывает, количество тактов, на которые сдвигается управляемый генератор.

Аргументами функция f_d являются значения регистра сдвига с адресами $0, 1, 3, 7, 12, 20, 30, 44, 65, 80$.

Как было указано в [1] нелинейность N_{f_d} функции f_d, используемой в LILI-128, равна 480.

Из этого следует, что существует такая линейная функция $f_l(x_1, \ldots, x_{10}) = a_1x_1 + a_2x_2 + \cdots + a_{10}x_{10}$, что $d_H(f, l) = 480$.

Таким образом, вероятность того, что значения функций f_d и f_l совпадут при одинаковых значениях аргументов равна:

$$P(f_d(x) = f_l(x)) = \frac{1024 - 480}{1024} = 0.53125.$$

Полный спектр Уолша-Адамара функции f_d следующий: существует 720 различных $\omega \in \mathbb{F}_2^{10}$ с $F(\omega) = 0$, 64 с $F(\omega) \pm 32$ и 240 с $F(\omega) \pm 64$.

Следовательно, существует в точности 240 различных аффинных функций $f_{l_1}, f_{l_2}, \ldots, f_{l_{240}}$, таких что

$$P(f_d(x) = f_{l_i}(x)) = 0.53125, \qquad 1 \leq i \leq 240.$$

[1] Fredrik Jönsson и Thomas Johansson (2002). “A fast correlation attack on LILI-128.”. В: *Inf. Process. Lett.* 81.3, с. 127—132. URL: http://dblp.uni-trier.de/db/journals/ipl/ipl81.html#JonssonJ02

В статье [2] предлагается быстрая корреляционной атака на алгоритм LILI-128, которая является развитием идей статьи [3].

Пусть N — количество знаков гаммы, вырабатываемых поточным шифром LILI-128.

Пусть число M — количество вырабатываемых знаков управляемым узлом ($M \approx 2.5N$).

---

[2] Fredrik Jönsson и Thomas Johansson (2002). “A fast correlation attack on LILI-128.”. В: *Inf. Process. Lett.* 81.3, с. 127—132. URL: http://dblp.uni-trier.de/db/journals/ipl/ipl81.html#JonssonJ02

[3] Vladimir V. Chepyzhov и др. (2000). “A Simple Algorithm for Fast Correlation Attacks on Stream Ciphers.”. В: *FSE*. Под ред. Bruce Schneier. Т. 1978. Lecture Notes in Computer Science. Springer, с. 181—195. URL: http://dblp.uni-trier.de/db/conf/fse/fse2000.html#ChepyzhovJS00

Обозначим последовательность, вырабатываемую функцией f_d, как $d = (d_1, d_2, \ldots d_M)$ и определим последовательность $s = (s_1, s_2, \ldots s_N)$, $s_k \in \mathbb{Z}$, где

$$s_k = \sum_{i=1}^{k} c_i, \quad k = 1, \ldots, N.$$

Используя последовательности d и s, определим последовательность z следующим образом:

$$z_k = d_{s_k}, \quad k = 1, \ldots, N,$$

где $N < M$.

Пусть опробовано начальное заполнение РСЛОС$_c$, что дает нам возможность построить последовательность z_k. Приведем краткое описание предварительного этапа восстановления ключа алгоритма LILI-128:

1. Определить 240 матриц G_i размером $240 \times N$, так, что $v_i = u_0 \times G_i$, где v_i — выходная последовательность аффинной функции f_{l_i}, u_0 — начальное состояние РСЛОС$_d$.
2. С помощью операции конкатенации составить матрицу $G' = (G_1 G_2 \dots G_N)$ размером $89 \times 240N$. Пусть $\mathbf{g}_i$ - i-ый столбец матрицы G', в таком случае:

$$G' = (\mathbf{g}_1\, \mathbf{g}_2 \dots \mathbf{g}_{240N}).$$

3. Найти тройки векторов, удовлетворяющих равенству

$$(\mathbf{g}_{i_1} + \mathbf{g}_{i_2} + \mathbf{g}_{i_3})^T = (\underbrace{*, *, \dots, *}_{k}, \underbrace{0, 0, \dots, 0}_{89\text{ - }k}), \tag{1}$$

где символ $*$ обозначает произвольное значение.

Обозначим число таких троек как m, а их индексы как

$$\{i_1(k), i_2(k), i_3(k)\},\ 1 \le k \le m.$$

В таком случае сумма $v_{i_1} + v_{i_2} + v_{i_3}$ есть линейная комбинация только первых k бит начального состояния РСЛОС$_d$. Тогда следующий набор

$$(v_{i_1(1)} + v_{i_2(1)} + v_{i_3(1)}, v_{i_1(2)} + v_{i_2(2)} + v_{i_3(2)}, \ldots, v_{i_1(m)} + v_{i_2(m)} + v_{i_3(m)})$$

есть слова линейного (N, k)-кода, который обозначим $\mathcal{C}_3$.

Пусть Z_k есть:

$$Z_k = z_{i_1(k)} + z_{i_2(k)} + z_{i_3(k)} \quad 1 \le k \le m,$$

тогда, вектор $(Z_1, Z_2, \ldots, Z_m)$ — принятое слово для $\mathcal{C}_3$.

Определим вектор ошибок для $\mathcal{C}_3$ как

$$(E_1, E_2, \ldots, E_m),$$

где

$$E_k = e_{i_1(k)} + e_{i_2(k)} + e_{i_3(k)}, \quad 1 \leq k \leq m$$

Считая e_i независимыми случайными величинами, то E_k при $k = 1, 2, \ldots, m$ также являются Бернуллиевскими случайными величинами с вероятностью 1 равной

$$p_3 = P(e_{i_1(k)} + e_{i_2(k)} + e_{i_3(k)} = 1) = 1/2 - 4\epsilon^3, \text{ где } \epsilon = 0.03125.$$

Для восстановления начального заполнения в работе [4] предлагается выполнить:

4. Декодирование принятого слова с использованием метода максимального правдоподобия. На выходе мы получаем кодовое слово, которое наиболее близко к $(Z_1, Z_2, \ldots, Z_m)$. Таким образом, мы получаем возможные k бит начального состояния РСЛОС$_d$. Оставшиеся $89-k$ бит начального состояния определяются таким же способом.

[4] Jönsson и Johansson 2002

Пусть k — натуральное число, удовлетворяющее условию: $k < 89$. На предварительном этапе необходимо найти все возможные тройки столбцов $\mathbf{g}_{i_1}, \mathbf{g}_{i_2}, \mathbf{g}_{i_3}$, удовлетворяющие условию (1), фактически определяющие линеаризующие соотношения для знаков гаммы. Для нахождения всех троек необходимо: представить столбы матрицы G' в виде списка, отсортированного по значению последних $89 - k$ бит.

Для каждой пары столбцов $\mathbf{g}_{i_1}, \mathbf{g}_{i_2}$ вычислить $\mathbf{g}_{i_1} + \mathbf{g}_{i_2}$ на последних $89 - k$ позициях. Далее для этой пары проводится проверка существования такого столбца $\mathbf{g_{i_3}}$, удовлетворяющего (1). Согласно Jönsson и Johansson 2002 вычислительная сложность предварительного этапа оценена величиной $O(N^2)$.

Авторами атаки предлагаются следующие оценки трудоемкости и необходимого количества материала для декодирования начального состояния РСЛОС$_d$.

Число троек векторов $(\mathbf{g}_{i_1} + \mathbf{g}_{i_2} + \mathbf{g}_{i_3})$, удовлетворяющих равенству (1), в среднем равно

$$m = \frac{(240N)^3}{3!} \cdot 2^{-89+k}.$$

Количество необходимого материала зависит от k и оценивается величиной:

$$N \approx \frac{1}{960} \cdot (k \cdot 12 \cdot \ln 2)^{1/3} \cdot \epsilon^{-2} \cdot 2^{\frac{89-k}{3}}.$$

Сложность декодирования C_{dec} оценена следующим выражением:

$$C_{dec} = 2^{k-5} \cdot k \cdot \frac{\ln 2}{\epsilon^6}.$$

Авторы заявляют, что трудоемкость предложенного метода анализа составляет $2^{39} \cdot C_{dec}$, где первый сомножитель отвечает за опробование начального заполнения РСЛОС$_c$. Эта формула не совсем корректна, так как:

1. не учитывается трудоемкость построения линеаризующих соотношений;
2. учитывается только трудоемкость восстановления k бит РСЛОС$_d$ и указывается, что «оставшиеся биты восстанавливаются аналогично и это имеет пренебрежимо малую трудоемкость».

Также заметим, что декодирование с использованием метода максимального правдоподобия всегда возвращает «наиболее вероятный» вектор, таким образом в работе не описан способ отбраковывания ложных значений РСЛОС$_c$.

Более того, в работе не оценивается вероятность успеха предложенного метода. В работе [5], на которую ссылаются авторы и где приводится описание быстрой корреляционной атаки, используемой в статье, без доказательств, на основе лишь эвристических соображений, утверждается, что вероятность успешного декодирования больше 1/2:

> *Simulations show that the critical length* $N = n_0$<..> *provides the probability of successful decoding close to* $1/2$*, while for* $N = 2n_0$ *the probability is close to* 1.

При этом для оценок трудоемкости, которые также используются в рассматриваемом методе анализа, используются только необходимые, а не достаточные условия однозначного декодирования, определяемые второй теоремой Шеннона.

[5] Vladimir V. Chepyzhov и др. (2000). "A Simple Algorithm for Fast Correlation Attacks on Stream Ciphers.". В: *FSE*. Под ред. Bruce Schneier. Т. 1978. Lecture Notes in Computer Science. Springer, с. 181—195. URL: http://dblp.uni-trier.de/db/conf/fse/fse2000.html#ChepyzhovJS00

Заметим, что если оставшиеся $89 - k$ бит восстанавливать «аналогичным способом», то, даже если верить эмпирическим оценкам вероятности, итоговая вероятность успеха метода будет не выше $2^{\left[\frac{89}{k}\right]}$, где $[x]$ — целая часть числа $x \in \mathbb{R}$, при этом трудоемкость возрастет не менее, чем в $\left[\frac{89}{k}\right]$ раз.

Если же восстанавливать оставшиеся значения полным опробованием, то трудоемкость увеличится на $2^{39} \cdot 2^{89-k}$, что больше заявляемой в работе трудоемкости определения ключа алгоритма LILI-128.

Авторы [6] утверждают, что предложенный в их работе метод анализа — суть статистический метод анализа, рассматриваемый в работе [7]. Оценим трудоемкость и вероятность восстановления ключа алгоритма LILI-128 с использованием описанного в работе Siegenthaler 1985, внеся дополнительные модификации, необходимые для применения метода для алгоритма LILI-128 и сравним полученные оценки.

Пусть α и β, соответственно, ошибки первого и второго рода. Тогда $(1 - \alpha)$ — вероятность найти ключ, β — вероятность принять ложный ключ за истинный.

[6] Vladimir V. Chepyzhov и др. (2000). “A Simple Algorithm for Fast Correlation Attacks on Stream Ciphers.”. В: *FSE*. Под ред. Bruce Schneier. Т. 1978. Lecture Notes in Computer Science. Springer, с. 181—195. URL: http://dblp.uni-trier.de/db/conf/fse/fse2000.html#ChepyzhovJS00

[7] Siegenthaler (1985). “Decrypting a Class of Stream Ciphers Using Ciphertext Only”. В: *IEEE Transactions on Computers* C-34.1, с. 81—85

Фактически изменяется только 4 шаг, который заменяется на следующие шаги:

4. Для каждого $k_1 \in K_1$, где $K_1 = \{0,1\}^k$ вычислим значение статистики S_n:

$$S_n = \frac{\sum_{j=1}^{m} (-1)^{z_{i_1(j)} + z_{i_1(j)} + z_{i_1(j)} + k_1' \times g_{i_1(j)} + k_1' \times g_{i_2(j)} + k_1' \times g_{i_3(j)}}}{\sqrt{m}},$$

   где k_1' есть конкатенация k_1 и последовательности из $89 - k$ нулей.

5. **Если** $S_n > c_\alpha$, где c_α — параметр метода, то для каждого $k_2 \in K_2, K_2 = \{0,1\}^{89-k}$, отбраковать ключ $k = k_1 \times k_2$, **иначе** k_1 - ложный.

Статистика S_n, используемая выше эквивалентна линейному члену статистики отношения правдоподобий.

При этом, в случае опробования истинного значения части ключа статистика будет иметь асимптотически нормальное распределение $\mathcal{N}(\delta\sqrt{m}, 1)$, где $\delta = 2\epsilon^3$, а в случае опробования ложного значения — $\mathcal{N}(0, 1)$.

Средняя трудоемкость статистического метода равна:

$$T_{st} \approx 2^k m + \beta 2^{89}.$$

Данную оценку трудоемкости можно уменьшить, используя в вычислениях быстрое преобразование Фурье (БПФ). При использовании БПФ трудоемкость $2^k \cdot m$ уменьшается до

$$\log_2(2^k) \cdot 2^k + m = k \cdot 2^k + m.$$

Следовательно, средняя трудоемкость статистического метода с использованием БПФ оценивается величиной:

$$T_{st} = k \cdot 2^k + m + \beta \cdot 2^{89}. \quad (2)$$

Наиболее корректной оценкой трудоемкости восстановления ключа алгоритма LILI-128 представленным выше методом с вероятностью успеха $1 - \alpha$ будет являться следующая величина:

$$T_{pr} + 2^{39} \cdot (D + T_{st}) + (1 - \alpha) \cdot 2^{89-k}, \tag{3}$$

где $T_{pr} = M^2$,

$$D = \frac{(240N)^3}{3!} \cdot 2^{-89+k}.$$

Заметим, что в отличие от (Jönsson и Johansson 2002) мы предлагаем на предварительном этапе сформировать таблицы объема $D \cdot k$ бит, а на этапе восстановления из D значений отобрать m подходящих, для которых реализовать статистический метод опробования. Данный подход очевидно позволяет сократить трудоемкость последних шагов алгоритма.

Пересчет оригинальной работы:

k	N	C_{dec}	$T_{fca} = C_{dec} + T_{pr}$	α	β
1	$2^{30.5}$	$2^{25.5}$	$2^{63.5}$	≈ 1	$3.8 \cdot 10^{-20}$
3	$2^{30.3}$	$2^{29.1}$	$2^{63.3}$	≈ 1	$7.9 \cdot 10^{-19}$
5	$2^{29.9}$	$2^{31.8}$	$2^{62.4}$	≈ 1	$3.3 \cdot 10^{-17}$
7	$2^{29.4}$	$2^{34.3}$	$2^{61.4}$	≈ 1	$3.3 \cdot 10^{-17}$
10	$2^{28.6}$	$2^{37.8}$	$2^{59.7}$	≈ 1	$3.8 \cdot 10^{-16}$
15	$2^{27.1}$	$2^{43.4}$	$2^{56.8}$	0.99	$1.9 \cdot 10^{-14}$
20	$2^{25.6}$	$2^{48.8}$	$2^{53.8}$	0.96	$7.9 \cdot 10^{-13}$
25	$2^{24.0}$	$2^{54.1}$	$2^{54.3}$	0.74	$3.2 \cdot 10^{-11}$

Отсюда можно сделать вывод, что при указанных в работе оценок на количество материала вероятность нахождения ключа близка к 0.

Рассмотрим формулу (3). Найдем минимальную трудоемкость при ошибке первого рода равной одному из следующих значений

$$\{0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.05, 0.1,$$
$$0.25, 0.5, 0.75, 0.8, 0.99, 0.999, 0.9999, 0.99999, 0.999999\},$$

а также найдем величину практической стойкости. Для нахождения значения трудоемкости воспользуемся оценкой (3), а для нахождения величины практической стойкости согласно (*ПНСТ 799-2022 Информационные технологии (ИТ). Криптографическая защита информации. Термины и определения* 2022) поделим величину трудоемкости на вероятность успеха, равную $1 - \alpha$. Для этого с использованием ЭВМ будем перебирать значения $N \in \{2^5, 2^6, 2^7, \ldots, 2^{50}\}$, $k \in \{1, 2, 3, \ldots, 88\}$.

Минимальная трудоемкость равняется $2^{75.19}$, при $k = 16$, $N = 2^{28}$, $1 - \alpha = 10^{-6}$. Отсюда $T_{pr} = 2^{58.64}$, $D = 2^{36.10}$, $T_{st} = 2^{32.13}$, $(1 - \alpha) \cdot 2^{89-k} = 2^{53.07}$.

Минимальная практическая стойкость относительно рассматриваемого метода равняется $2^{78.19}$, при $k = 21$, $N = 2^{27}$, $\alpha = 0.5$, откуда $T_{pr} = 2^{77.19}$, $D = 2^{38.10}$, $T_{st} = 2^{34.14}$, $(1 - \alpha) \cdot 2^{89-k} = 2^{67}$.

Заметим, что эта величина соответствует оценке, приведенной в оригинальной работе, однако в указанной работе не была оценена вероятность успеха и данная величина была получена из других соображений и не вполне обоснована:

> *As an example <..>, we choose* $k = 5$ *to get the complexity to approximately* 2^{32}. *Since this decoding phase has to be repeated for each possible initial state of* $LFSR_c$, *we get that the average complexity is* $2^{32} \cdot 2^{39} = 2^{71}$ *<..> The precomputation for this case is approximately* 2^{79}.

Заметим, что при параметрах метода, рассмотренных в статье (Jönsson и Johansson 2002) трудоемкость будет сравнима с трудоемкостью тотального опробования.

Thank you for your attention!

Questions?