Труды научно–технической конференции кластера пензенских предприятий,
обеспечивающих **БЕЗОПАСНОСТЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ.**
Том 10, Пенза–2016, с. 92 – 95 (http://пниэи.рф/activity/science/BIT/T10–p92.pdf)

УДК: 519.2, 612.087

Акмаев А.Ж. (г. Пенза)

# Об одном подходе к построению масштабируемой скоростной оптимизации криптографического алгоритма ГОСТ Р 34.12-2015 с длиной блока 128 бит за счет использования предвычисленных таблиц

## Введение

В 2015 году в Российской Федерации был принят новый стандарт симметричного блочного криптографического алгоритма – ГОСТ Р 34.12-2015[1]. В тексте стандарта приводится описание двух принципиально разных криптографических алгоритмов с длинами блока 64 бит («Магма») и 128 бит («Кузнечик»). Длина ключа для обоих алгоритмов составляет 256 бит.

Криптографический алгоритм «Магма» построен на основе сети Фейстеля и является по сути ребрендингом алгоритма ГОСТ 28147-89[2] с фиксированными узлами замены и отличными режимами работы.

Криптографический алгоритм «Кузнечик», в свою очередь, представляет собой совершенно новый алгоритм, построенный с использованием современного математического аппарата и последних достижений в области криптографии. В основе «Кузнечика» лежит так называемой LSX-преобразование, а в качестве базовой структуры алгоритма используется подстановочно-перестановочная сеть (SP-сеть).

По состоянию на начало 2017 года известны варианты [7] скоростной оптимизации криптографического алгоритма «Кузнечик» за счет использования предвычисленных таблиц. При этом существующие подходы к оптимизации используют предвычисленные таблицы объемом 64 кБайт. Безусловно, данные варианты оптимизации удобны для программной реализации на современных процессорах общего назначения [4]. Однако, некоторые задачи, решаемые специалистами АО «ПНИЭИ», требуют одновременно, как жесткую экономию памяти (к примеру, на сигнальных микропроцессорах), так и ускорение вычислений. В связи, с этим был разработан подход к построению масштабируемой скоростной реализации алгоритма ГОСТ Р 34.12-2015. Выбор коэффициента оптимизации зависит от доступного разработчику программного обеспечения объема памяти для хранения предвычисленных таблиц.

## Описание криптографического алгоритма «Кузнечик»

Ниже приводится сжатое описание криптографического алгоритма «Кузнечик» в формализованном виде. На рисунке 1 приведена схема одного такта работы алгоритма, т.е. последовательных преобразований *LSX*.

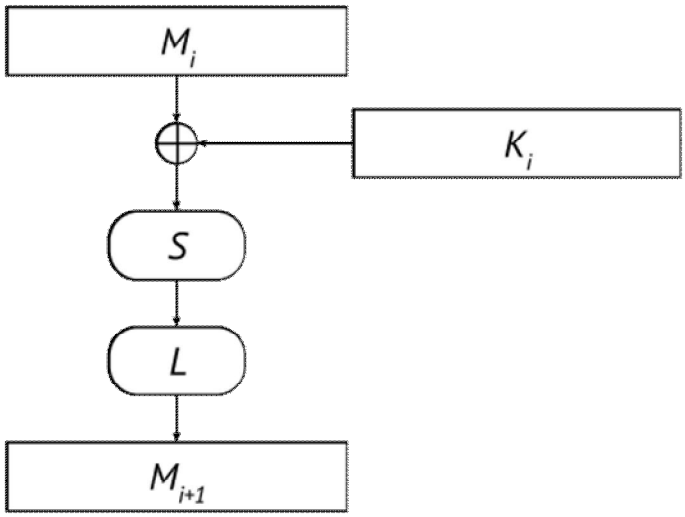


Рис. 1 – Такт работы алгоритма «Кузнечик»

В формализованном виде процедуру зашифрования можно записать следующим образом

$$E_{K_1 \dots K_{10}}(a) = X\left[K_{10}\right] LSX\left[K_9\right] \dots LSX\left[K_2\right] LSX\left[K_1\right](a).$$

| | |
|---|---|
| $X[k]: V_{128} \to V_{128}$ | $X[k](a) = k \oplus a$, где $k, a \in V_{128}$ |
| $\pi: V_8 \to V_8$ | $\pi(a) = b$, где $a, b \in V_8$ |
| $S: V_{128} \to V_{128}$ | $S(a) = S(a_{15} \| \dots \| a_0) = \pi(a_{15}) \| \dots \| \pi(a_0)$, где $a = a_{15} \| \dots \| a_0 \in V_{128}, a_i \in V_8, i = 0,1,\dots,15$ |
| $l: V_{128} \to V_8$ | $l(a_{15},\dots,a_0) = \bigoplus_{i=0}^{15} k_i \circ a_i$, где $k_i, a_i \in V_8$. При этом коэффициенты $k_i$ являются фиксированными, а операция умножения осуществляется в поле $GF(2)[x] / p(x)$, где $p(x) = x^8 + x^7 + x^6 + x + 1 \in GF(2)[x]$. |
| $R: V_{128} \to V_{128}$ | $R(a) = R(a_{15} \| \dots \| a_0) = l(a_{15},\dots,a_0) \| a_{15} \| \dots \| a_1$, где $a = a_{15} \| \dots \| a_0 \in V_{128}, a_i \in V_8, i = 0,1,\dots,15$ |
| $L: V_{128} \to V_{128}$ | $L(a) = R^{16}(a)$, где $a \in V_{128}$ |

Следует отметить, что преобразование *R* представляет собой по сути 1 такт работы регистра сдвига с линейной обратной связью (РСЛОС) заданного над элементами поля $GF(2)[x] / p(x)$, а преобразование *L* – соответственно, реализация 16 последовательных тактов работы регистра.

В конце 90-ых годов было установлено [3], что для шифров семейства *LSX,* в качестве рассеивающего преобразования, наиболее оптимальным с точки зрения распространения ошибок (для противодействия линейному и дифференциальному криптоанализу) является использование так называемых MDS-матриц [3,5,6]. Создатели алгоритма «Кузнечик» изящно подошли к построению рассеивающего преобразования, использовав в его качестве матрицу рекурсивного МДР кода.

Очевидно, что запись преобразования $L$ в виде регистра сдвига с заданным полиномом или в виде матричного преобразования, аналогичны. Таким образом, запись в виде преобразований РСЛОС

$$L(a) = R^{16}(a) = R^8R^8(a) = R^4R^4R^4R^4(a) = R^2R^2R^2R^2R^2R^2R^2R^2(a)$$

можно заменить на запись в виде умножения вектора на матрицу

$$\mathbf{a} \times \mathbf{L} = \mathbf{a} \times \mathbf{R}^{16} = \mathbf{a} \times \mathbf{R}^8 \times \mathbf{R}^8.$$

**Построение оптимизационных решений**

С момента появления криптографического алгоритма *AES* известны оптимизации (за счет использования предвычисленных таблиц) умножения вектора длины $n$ на матрицу длины $n \times n$, позволяющие реализовать данную операцию со сложностью $O(n)$ вместо $O(n^2)$.

В 2014 году специалистами ОАО «ИнфоТеКС» была предложена [7] оптимизация криптографического алгоритма «Кузнечик», использующая две предвычисленные таблицы размером 64 кБайта для процедуры зашифрования и расшифрования соответственно.

Однако, при решении практических задачах зачастую возникает ситуация так называемых торгов «время/память», равновесная точка которых может меняться в зависимости от доступных программисту ресурсов памяти, а также требуемых скоростных характеристик реализации.

Для построения масштабируемой оптимизации алгоритма «Кузнечик» рассмотрим следующие особенности преобразования $L$ и процедуры умножения вектора на фиксированную матрицу.

Пусть даны вектор-строка **a** размера $n$ и матрица $\mathbf{L}$ размера $n \times n$, тогда результат их умножения **b** представляется в виде $\mathbf{b} = (b_0, \ldots, b_{n-1})$, где $b_i = \bigoplus_{j=0}^{n-1} a_j l_{j,i}$. Очевидно, сложность умножения вектора на матрицу при последовательном умножении и сложении каждого элемента, составляет $O(n^2)$ операций. Однако, вектор **b** может быть записан в виде $\mathbf{b} = (b_0, \ldots, b_{n-1}) = (\bigoplus_{j=0}^{n-1} a_j l_{j,0}, \ldots, \bigoplus_{j=0}^{n-1} a_j l_{j,n-1}) = \bigoplus_{j=0}^{n-1} a_j (l_{j,0}, \ldots, l_{j,n-1})$. Данная запись означает, что при использовании предвычисленных таблиц вида $a_j (l_{j,0}, \ldots, l_{j,n-1})$ можно сократить сложность вычисления результата умножения вектора на фиксированную матрицу до $O(n)$ операций. Таким образом, используя 16

таблиц размером 256 на 16 байт можно заменить «сложную» операцию вычисления элементов в поле на табличные замены. Кроме того, при такой реализации существует принципиальная возможность объединения операций *LS* и хранение в предвычисленных таблицах результата их последовательного применения, т.е. $Tab_j[a_j] = \pi(a_j)(l_{j,0},\ldots,l_{j,n-1})$.

К сожалению, из-за жестких ограничений на объем используемой памяти не всегда возможно обеспечение хранения предвычисленных таблиц такого объема. В связи с этим, учитывая структурные особенности матрицы, реализующей РСЛОС, была предложена следующая декомпозиция преобразования *L*

$$L(a_0 \| \ldots \| a_{15}) = R^8(R^8(a)) = T(a_0 \| \ldots \| a_{15}) \| T(a_0 \| \ldots \| a_{15} \| T(a_0 \| \ldots \| a_{15})),$$

где $T : V_{128} \to V_{64}$.

$$T(a_0 \| \ldots \| a_{15}) = \bigoplus_{i=0}^{15} Tab_i[a_i], \text{ где } a_i \in V_8.$$

Для данной реализации необходимо 16 таблиц размером 256 на 8 байт, кроме того, требуется в 2 раза больше табличных замен, а также отсутствует возможность табличного объединения операции *LS*.

Очевидно, что используя предложенный подход возможно уменьшение таблиц еще в 2 и 4 раза, реализующих матрицы $R^4$ и $R^2$ соответственно.

**Выводы**

В статье предложен подход к построению масштабируемой скоростной оптимизации криптографического алгоритма ГОСТ Р 34.12-2015 с длиной блока 128 бит за счет использования предвычисленных таблиц.

**Литература:**


1) ГОСТ Р 34.12-2015 «Информационные технологии. Криптографическая защита информации. Блочные шифры».

2) ГОСТ 28147-89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования».

3) The design of Rijndael: AES - The Advanced Encryption Standard. Joan Daemen, Vincent Rijmen.

4) Об эксплуатационных качествах одного перспективного блочного шифра типа LSX. Е. К. Алексеев, В. О. Попов, А. С. Прохоров, С. В. Смышляев, Л. А. Сонина. МАТЕМАТИЧЕСКИЕ ВОПРОСЫ КРИПТОГРАФИИ 2015 Т. 6 № 2 С. 7–17.



Статья поступила 10.12.2016, опубликована 27.12 .2016 по положительной рецензии д.т.н. Малыгина А.Ю.