О глубине аппаратной реализации блочного шифра Кузнечик

Е. А. Курганов

В работе исследуется глубина аппаратной (схемной) реализации блочного шифра Кузнечик. Также проводится сравнение глубины реализации алгоритма Кузнечик с другими распространенными алгоритмами блочного шифрования.

Ключевые слова: оптимизация глубины схем, блочные шифры.

Введение

С 1 января 2016 года в РФ введен новый стандарт блочного шифрования ГОСТ 34.12-2015 [1], который заменил ГОСТ 28147-89 [2]. Данный шифр разработан Центром защиты информации и специальной связи ФСБ России с участием ОАО «ИнфоТеКС». Если раньше в Государственный стандарт входил только один алгоритм шифрования для блоков размером 64 бита, то новый стандарт включает в себя 2 алгоритма — для блоков размером 128 и 64 бита. Эти алгоритмы получили неофициальные названия Кузнечик и Магма соответственно. Причем алгоритм Магма — тот же самый, что и в ГОСТ 28147-89 с той лишь разницей, что теперь для него зафиксированы блоки подстановок (см. [1]).

В данной статье речь пойдет об аппаратной реализации нового алгоритма. Определяющим параметром производительности

таких реализаций является глубина схемы. Под глубиной понимается длина максимального простого пути схемы. Рассматривается базис из элементов конъюнкции, дизъюнкции, отрицания и задержки. При этом отрицание игнорируется при вычислении глубины (вычисление проводится по тем же правилам, что и в [3]).

В начале статьи подробно описываются алгоритмы Кузнечик и Магма. Далее кратко описываются алгоритмы AES, DES, 3DES. Затем приводится оценка глубины для всех преобразований, используемых в этих алгоритмах. В конце статьи вычисляется глубина раунда и общая глубина для каждого алгоритма.

Алгоритм Кузнечик

Входными данными для данного алгоритма являются блок открытого текста размером 128 бит и ключ размером 256 бит. По своей структуре новый шифр — это SP-сеть (т. е. преобразование производится со всем блоком данных, а не с половиной), которая состоит из 9 одинаковых раундов и еще одного дополнительного сложения с раундовым ключом. Один раунд данной сети имеет следующую структуру:

1) побитовое сложение (XOR) с раундовым ключом K_i;

2) нелинейное преобразование блоком подстановок S;

3) линейное преобразование L.

Схема одного раунда шифра Кузнечик представлена на рисунке 1. Общая схема шифра — на рисунке 3.

Нелинейное преобразование

Нелинейное преобразование S представляет собой применение к каждому 8-битному подвектору 128-битного входного вектора фиксированной подстановки π, описание которой можно найти в [1]:

$$S(a) = S(a_{15}|| \dots ||a_0) = \pi(a_{15})|| \dots ||\pi(a_0),$$

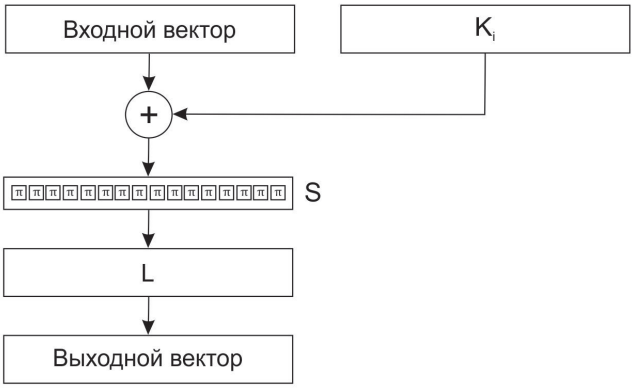

Рис. 1: Структура раунда алгоритма Кузнечик.

где a — 128-битный вектор. Та же самая подстановка π используется и в хэш-функции Стрибог [4]. В работе [5] показано, как устроена эта подстановка.

Линейное преобразование

Линейное преобразование L определяется следующим образом:

$$L(a) = R^{16}(a);$$
$$R(a) = R(a_{15}\|a_{14}\|\ldots\|a_0) = \ell(a_{15},\ldots,a_0)\|a_{15}\|\ldots\|a_1;$$
$$\ell(a_{15},\ldots,a_0) = \nabla(148\cdot\Delta(a_{15}) + 32\cdot\Delta(a_{14}) + 133\cdot\Delta(a_{13}) +$$
$$16\cdot\Delta(a_{12}) + 194\cdot\Delta(a_{11}) + 192\cdot\Delta(a_{10}) + 1\cdot\Delta(a_9) + 251\cdot\Delta(a_8) +$$
$$1\cdot\Delta(a_7) + 192\cdot\Delta(a_6) + 194\cdot\Delta(a_5) + 16\cdot\Delta(a_4) + 133\cdot\Delta(a_3) +$$
$$32\cdot\Delta(a_2) + 148\cdot\Delta(a_1) + 1\cdot\Delta(a_0)),$$

где

- $\Delta : V_8 \to \mathbb{F}$ — биективное отображение, ставящее в соответствие двоичной строке элемент поля $\mathbb{F}$;
- $\nabla : \mathbb{F} \to V_8$ — отображение, обратное к Δ;
- $\mathbb{F}$ — поле Галуа $GF(2^8)$ с неприводимым многочленом $x^8 + x^7 + x^6 + x + 1$.

Как видно, преобразование L может быть реализовано с помощью линейного регистра сдвига с обратной связью, который совершает 16 итераций (см рис. 2).

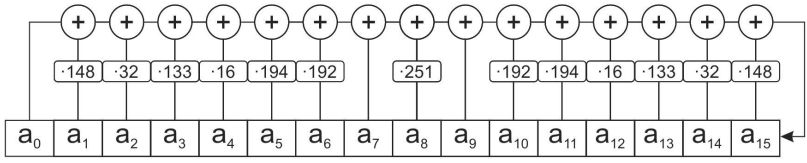

Рис. 2: Линейный регистр сдвига с обратной связью в преобразовании L.

Генерация раундовых ключей

Первые два раундовых ключа получаются разбиением исходного ключа пополам. Для выработки следующих раундовых ключей используется сеть Фейстеля, нелинейное преобразование которой — функция LSX, которая используется в самом шифре. Раундовым ключом в данном случае служит номер итерации данной сети, к которому применяется преобразование L. Каждые 8 итераций данной сети получается новая пара раундовых ключей. Поэтому всего у нее 32 итерации. Общая схема генерации раундовых ключей изображена на рисунке 4.

Стоит отметить, что при шифровании нескольких блоков информации развертка раундовых ключей производится только один раз, после чего используются ключи, полученные ранее. Поэтому данную процедуру можно игнорировать при вычислении глубины шифра.

Алгоритм Магма

Входными данными для алгоритма Магма являются блок открытого текста размером 64 бит и ключ размером 256 бит. По своей структуре данный шифр — это сеть Фейстеля, которая совершает 32 итерации. Нелинейная функция данного алгоритма устроена следующим образом:

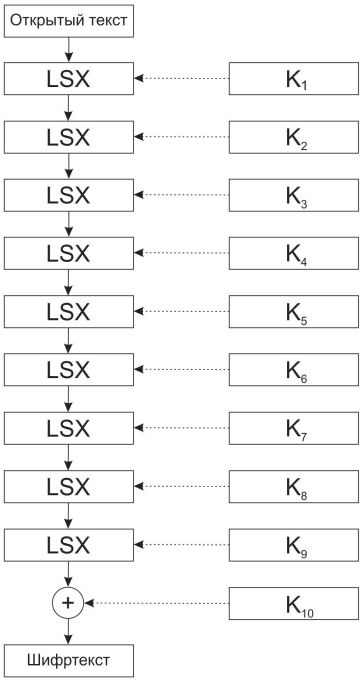

Рис. 3: Общая структура алгоритма Кузнечик.

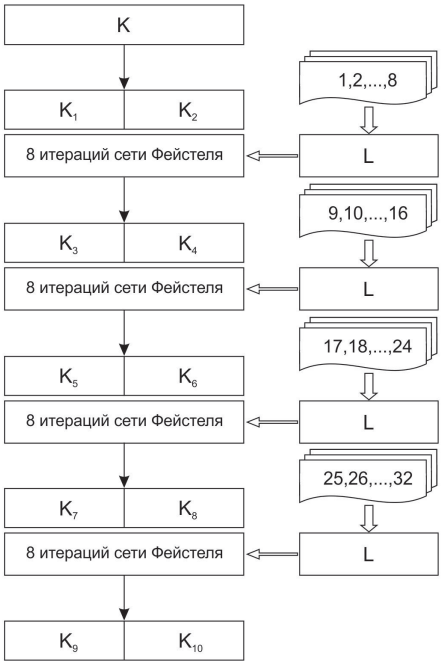

Рис. 4: Генерация раундовых ключей.

1) сложение с раундовым ключом K_i по модулю 2^{32};

2) каждый 4-битный подвектор 32-битного входного вектора проходит через S-блок S_i;

3) полученный 32-битный вектор циклически сдвигается влево на 11 бит.

Общая структура данного алгоритма изображена на рисунке 5, структура раунда — на рисунке 6.

Генерация раундовых ключей

Все раундовые ключи K_i вырабатываются на основе исходного ключа K согласно следующим равенствам:

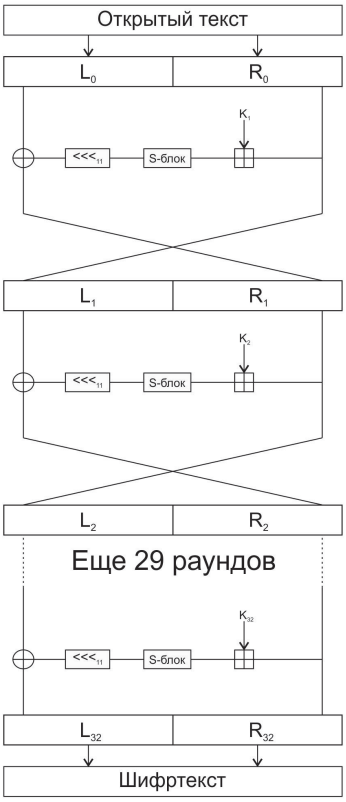

Рис. 5: Общая структура алгоритма Магма.

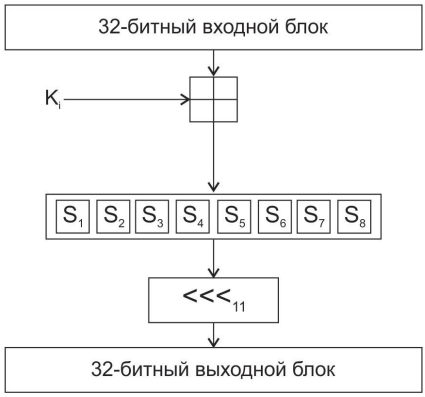

Рис. 6: Структура раунда алгоритма Магма.

$$
\begin{aligned}
&K = k_{255}\|k_{254}\|\ldots\|k_0;\\
&K_1 = k_{255}\|k_{254}\|\ldots\|k_{224};\\
&K_2 = k_{223}\|k_{222}\|\ldots\|k_{192};\\
&\ldots\\
&K_8 = k_{31}\|k_{30}\|\ldots\|k_0;\\
&K_{i+8} = K_i, \qquad i = 1,2,\ldots,8;\\
&K_{i+16} = K_i, \qquad i = 1,2,\ldots,8;\\
&K_{i+24} = K_{9-i}, \qquad i = 1,2,\ldots,8.
\end{aligned}
$$

Другие алгоритмы блочного шифрования

Алгоритм AES

AES (Advanced Encryption Standard) — алгоритм блочного шифрования, принятый в качестве стандарта шифрования в США [6]. Входными данными являются блок открытого текста размером 128 бит и ключ размером 128, 192 или 256 бит. В зависимости от длины ключа варьируется и число раундов алгоритма (10, 12 или 14 соответственно). Один раунд состоит из 4 преобразований:

- SubBytes — преобразование S-блоком 8×8 бит;
- ShiftRows — перестановка байтов во всем 128-битном блоке;
- MixColumns — умножение на матрицу слева в поле $GF(2^8)$ (в последнем раунде не производится);
- AddRoundKey — побитовое сложение с раундовым ключом.

Также в начале алгоритма производится сложение с раундовым ключом. Преобразование MixColumns можно записать в следующем виде:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} w_3 \\ w_2 \\ w_1 \\ w_0 \end{bmatrix} = \begin{bmatrix} w'_3 \\ w'_2 \\ w'_1 \\ w'_0 \end{bmatrix}$$

в качестве неприводимого многочлена в $GF(2^8)$ используется $x^8 + x^4 + x^3 + x + 1$.

Алгоритм DES

DES (Data Encryption Standard) — алгоритм блочного шифрования, ранее использовавшийся в качестве стандарта шифрования в США [7]. Входными данными являются блок открытого текста размером 64 бит и ключ размером 64 бит (используется

только 56). Число раундов данного алгоритма — 16. Один раунд состоит из 4 преобразований:

- функция расширения E (перестановка, 32 $\rightarrow$ 48 бит);
- побитовое сложение с ключом K_i;
- 8 S-блоков 6$\times$4 бита;
- перестановка P.

Также в начале и в конце алгоритма применяются две разные перестановки (подробнее см. [7]).

Алгоритм 3DES

Из-за того, что алгоритм DES имел маленькую длину ключа, был создан алгоритм 3DES [7]. Его идея заключается в том, что для шифрования сообщения надо последовательно три раза применить алгоритм DES с тремя ключами. Причем на каждом шаге используются либо 3 разных ключа, либо ключи на первом и третьем шаге совпадают.

Оценка глубины используемых преобразований

Линейное преобразование алгоритма Кузнечик

Непосредственной проверкой можно убедиться, что одну итерацию линейного регистра сдвига с обратной связью можно записать в виде произведения исходного вектора на данную матрицу слева (в поле $GF(2^8)$):

$$\begin{bmatrix}
94 & 20 & 85 & 10 & C2 & C0 & 01 & FB & 01 & C0 & C2 & 10 & 85 & 20 & 94 & 01 \\
01 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\
00 & 01 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\
00 & 00 & 01 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\
00 & 00 & 00 & 01 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\
00 & 00 & 00 & 00 & 01 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\
00 & 00 & 00 & 00 & 00 & 01 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\
00 & 00 & 00 & 00 & 00 & 00 & 01 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\
00 & 00 & 00 & 00 & 00 & 00 & 00 & 01 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\
00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 01 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\
00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 01 & 00 & 00 & 00 & 00 & 00 & 00 \\
00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 01 & 00 & 00 & 00 & 00 & 00 \\
00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 01 & 00 & 00 & 00 & 00 \\
00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 01 & 00 & 00 & 00 \\
00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 01 & 00 & 00 \\
00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 01 & 00
\end{bmatrix}$$

Следовательно, 16 итераций данного регистра можно записать в виде произведения исходного вектора на эту же матрицу, возведенную в 16 степень. В этом также можно убедиться, выполнив проверку. Получившаяся матрица приведена ниже.

$$\begin{bmatrix}
CF & 98 & 74 & BF & 93 & 8E & F2 & F3 & 0A & BF & F6 & A9 & EA & 8E & 4D & 6E \\
6E & 20 & C6 & DA & 90 & 48 & 89 & 9C & C1 & 64 & B8 & 2D & 86 & 44 & D0 & A2 \\
A2 & C8 & 87 & 70 & 68 & 43 & 1C & 2B & A1 & 63 & 30 & 6B & 9F & 30 & E3 & 76 \\
76 & 33 & 10 & 0C & 1C & 11 & D6 & 6A & A6 & D7 & F6 & 49 & 07 & 14 & E8 & 72 \\
72 & F2 & 6B & CA & 20 & EB & 02 & A4 & 8D & D4 & C4 & 01 & 65 & DD & 4C & 6C \\
6C & 76 & EC & 0C & C5 & BC & AF & 6E & A3 & E1 & 90 & 58 & 0E & 02 & C3 & 48 \\
48 & D5 & 62 & 17 & 06 & 2D & C4 & E7 & D5 & EB & 99 & 78 & 52 & F5 & 16 & 7A \\
7A & E6 & 4E & 1A & BB & 2E & F1 & BE & D4 & AF & 37 & B1 & D4 & 2A & 6E & B8 \\
B8 & 49 & 87 & 14 & CB & 8D & AB & 49 & 09 & 6C & 2A & 01 & 60 & 8E & 4B & 5D \\
5D & D4 & B8 & 2F & 8D & 12 & EE & F6 & 08 & 54 & 0F & F3 & 98 & C8 & 7F & 27 \\
27 & 9F & BE & 68 & 1A & 7C & AD & C9 & 84 & 2F & EB & FE & C6 & 48 & A2 & BD \\
BD & 95 & 5E & 30 & E9 & 60 & BF & 10 & EF & 39 & EC & 91 & 7F & 48 & 89 & 10 \\
10 & E9 & D0 & D9 & F3 & 94 & 3D & AF & 7B & FF & 64 & 91 & 52 & F8 & 0D & DD \\
DD & 99 & 75 & CA & 97 & 44 & 5A & E0 & 30 & A6 & 31 & D3 & DF & 48 & 64 & 84 \\
84 & 2D & 74 & 96 & 5D & 77 & 6F & DE & 54 & B4 & 8D & D1 & 44 & 3C & A5 & 94 \\
94 & 20 & 85 & 10 & C2 & C0 & 01 & FB & 01 & C0 & C2 & 10 & 85 & 20 & 94 & 01
\end{bmatrix}$$

Однако вычисление произведения двух чисел в поле Галуа выглядит довольно громоздко. Пусть сначала надо вычислить

произведение $a \cdot 2$, где $a = \{a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0\}$. Тогда

$$\begin{aligned} a \cdot 2 = &\{a_6, a_5, a_4, a_3, a_2, a_1, a_0, 0\} \\ &+ \{a_7, a_7, 0\ , 0\ , 0\ , 0\ , a_7, a_7\}. \end{aligned}$$

Очевидно, что это произведение можно вычислить с глубиной 2. Пусть теперь требуется вычислить произведение $a \cdot 2^i$, $i = 2, 3, \ldots, 7$. Для этого надо i раз повторить процедуру, описанную выше (см. рис. 7). Значит, максимальная глубина подобного вычисления будет $2 \cdot 7 = 14$. Подобная техника вычисления предложена в [8]. Для того, чтобы вычислить произведение a на произвольный элемент поля Галуа, нужно разложить его по степеням двойки, затем параллельно вычислить все необходимые произведения вида $a \cdot 2^i$, после чего побитово просуммировать их. В худшем случае глубина такого вычисления будет еще на 2 больше (если $a > 2^7$). И, наконец, чтобы вычислить произведение строки на вектор, требуется еще побитово просуммировать 16 чисел, что можно сделать с глубиной $2 \cdot \log_2 16 = 8$. Следовательно, при таком представлении глубина вычисления преобразования L не превосходит 24.

Для того, чтобы уменьшить глубину вычисления преобразования L, можно перейти от матрицы в поле $GF(2^8)$ размером 16×16 к матрице в поле $GF(2)$ размером 128×128. Сделать это можно, сопоставив каждому элементу поля Галуа из исходной матрицы матрицу из $GF(2)$ размером 8×8 по следующему правилу:

- элементу 0 соответствует матрица из всех нулей;
- элементу 1 соответствует единичная матрица;

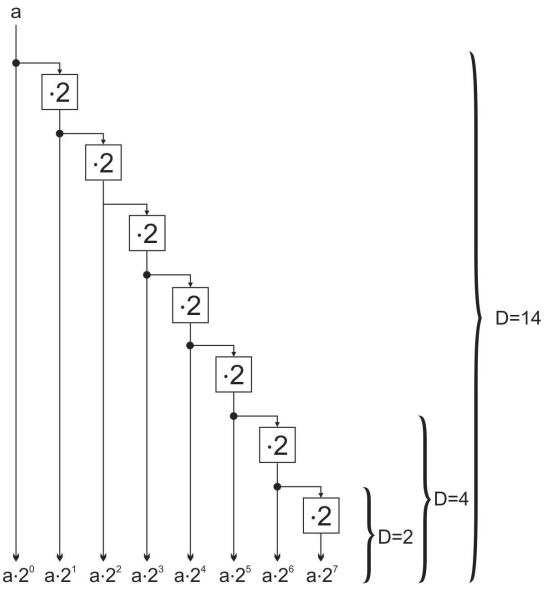

Рис. 7: Схема умножения двух чисел в поле $GF(2^8)$.

- элементу 2 соответствует следующая матрица A:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- элементу 2^i, $i = 2, 3, \ldots, 7$ соответствует матрица A^i;
- произвольному элементу $a = \{a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0\}$ соответствует сумма матриц A^i для всех i таких, что $a_i = 1$.

В справедливости данного соответствия можно убедиться непосредственной проверкой. Получившаяся таким образом матрица имеет максимум 74 единицы в одной строке, поэтому в

данную схему можно добавить еще 54 слагаемых без увеличения глубины. Таким образом, преобразование L можно вычислить с глубиной $\lceil \log_2 74 \rceil \cdot 2 = 14$.

S-блок

S-блок размером 8×8 бит можно реализовать, воспользовавшись СДНФ [9]. Для этого надо:

1) вычислить всевозможные произведения вида

$$x_1^{\sigma_1} \& x_2^{\sigma_2} \& x_3^{\sigma_2} \& x_4^{\sigma_2} \& x_5^{\sigma_2} \& x_6^{\sigma_2} \& x_7^{\sigma_2} \& x_8^{\sigma_2},$$

где $\sigma_i = 0, 1,\ i = 1, \ldots, 8;\quad x^1 = x,\ x^0 = \bar{x}$ (всего их будет 256, на каждом наборе только одно из них будет равно 1); это можно сделать с глубиной 3. Каждому такому произведению можно поставить в соответствие число от 0 до 255;

2) для каждого выхода $y_j,\ j = 1, \ldots, 8$ выбрать 128 из 256 получившихся на прошлом шаге произведений, которым соответствуют те входные значения, на которых $y_j = 1$;

3) для каждого выхода $y_j,\ j = 1, \ldots, 8$ посчитать дизъюнкцию 128 чисел, выбранных на прошлом шаге; это можно сделать с глубиной 7.

Общая глубина данной конструкции равна 10.

Подобным образом можно реализовать S-блоки размером 4×4 и 6×4 бита с глубиной 5 и 8 соответственно.

Сложение по модулю 2^{32}

Для того, чтобы вычислить сумму двух 32-битных векторов по модулю 2^{32}, можно воспользоваться методом золотого сечения, описанным в [10]. Тогда данную операцию можно реализовать с глубиной 10.

Преобразование MixColumns алгоритма AES

Данное преобразование легко реализуется с глубиной 6. Каждая строка матрицы данного преобразования состоит из одинаковых элементов: 01, 01, 02, 03. Поэтому произведение каждой строки матрицы на вектор можно вычислить с одинаковой глубиной. Распишем его подробнее:

$$w'_i = 01{\cdot}w_{i_0} \oplus 01{\cdot}w_{i_1} \oplus 02{\cdot}w_{i_2} \oplus 03{\cdot}w_{i_3} = w_{i_0} \oplus w_{i_1} \oplus 02{\cdot}w_{i_2} \oplus 02{\cdot}w_{i_3} \oplus w_{i_3}.$$

Ранее было показано, что умножение на 2 в поле Галуа можно произвести с глубиной 2. Поэтому глубина вычисления данного преобразования равна $\lceil \log_2 7 \rceil \cdot 2 = 6$.

Более подробно об этом можно прочитать в [8].

Сравнение глубины алгоритмов блочного шифрования

Воспользовавшись полученными результатами, можно вычислить глубину раунда для каждого алгоритма:

- Кузнечик. Один раунд состоит из линейного преобразования L, нелинейного преобразования S и побитового сложения с раундовым ключом. Поэтому
  $D \leqslant D(L) + D(S) + D(\oplus) = 14 + 10 + 2 = 26$.
  Однако сложение с раундовым ключом можно делать внутри линейного преобразования, начиная со второго раунда. Это позволит уменьшить глубину всего вычисления на 2. Следовательно, глубина одного раунда Кузнечика не превосходит 24.

- Магма. Один раунд состоит из сложения по модулю 2^{32}, нелинейного преобразования на основе S-блоков 4×4 бит и побитового сложения с левой половиной блока. Поэтому $D \leqslant D(\boxplus) + D(S) + D(\oplus) = 10 + 5 + 2 = 17$.

- AES. Один раунд состоит из преобразований SubBytes, MixColumns, ShiftRows и побитового сложения с раундовым ключом.

$D \leqslant D(SubBytes) + D(MixColumns) + D(\oplus) = 10 + 6 + 2 = 18$.
Однако здесь сложение с раундовым ключом можно выполнять во время преобразования MixColumns. Это уменьшит глубину вычисления на 2. Поэтому глубина раунда AES не превосходит 16.

- DES. Один раунд состоит из нелинейного преобразования на основе S-блоков 6×4 бит, побитового сложения с раундовым ключом и побитового сложения с левой половиной блока. Поэтому
$D \leqslant D(S) + D(\oplus) + D(\oplus) = 8 + 2 + 2 = 12$.

Теперь можно вычислить общую глубину:

- Кузнечик. Алгоритм состоит из 9 одинаковых раундов (в первом раунде надо дополнительно учесть глубину одного побитового сложения с раундовым ключом) и еще одного побитового сложения в конце алгоритма, которое можно выполнить внутри линейного преобразования последнего раунда. Поэтому
$D \leqslant 24 \cdot 9 + 2 = 218$.

- Магма. Алгоритм состоит из 32 одинаковых раундов. Поэтому
$D \leqslant 17 \cdot 32 = 544$.

- AES-256. В начале алгоритма производится сложение с раундовым ключом. Затем следует 13 одинаковых раундов и последний раунд, в котором не производится преобразование MixColumns. Следовательно,
$D \leqslant 2 + 16 \cdot 13 + 12 = 222$.

- DES. Алгоритм состоит из 16 одинаковых раундов. Отсюда
$D \leqslant 12 \cdot 16 = 192$.

- 3DES. Алгоритм состоит из 48 одинаковых раундов. Поэтому
$D \leqslant 12 \cdot 16 \cdot 3 = 576$.

Полученные данные сведены в таблицу.

Алгоритм	**Глубина раунда**	**Общая глубина**	**Ур. логики/бит**
Кузнечик	24	218	1.7
Магма	17	544	8.5
AES-256	16	222	1.73
DES	12	192	3.0
3DES	12	576	9.0

Данная таблица показывает, что Кузнечик имеет самую большую глубину одного раунда среди рассмотренных алгоритмов. Однако при этом его показатель количества уровней логики на бит минимален и практически совпадает с AES-256.

Автор выражает благодарность своему научному руководителю, с. н. с. Алексею Владимировичу Галатенко, за постановку задачи и внимание к работе.

Список литературы

[1] ГОСТ. Криптографическая защита информации. Блочные шифры. Государственный стандарт РФ. 2015.

[2] ГОСТ. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. Государственный стандарт СССР. 1989.

[3] Болотов А. А., Галатенко А. В., Гринчук М. И., Золотых А. А., Иванович Л. Методы оптимизации глубины реализации хэш-функций. Интеллектуальные системы. 2013. № 17:1-4. С. 224–228.

[4] ГОСТ. Криптографическая защита информации. Функция хэширования. Государственный стандарт РФ. 2012.

[5] Biryukov A., Perrin L., Udovenko A. The Secret Structure of the S-Box of Streebog, Kuznechik and Stribob. SnT, University of Luxembourg. 2015.

[6] NIST NIST, FIPS PUB 197. 2001.

[7] NIST NIST, FIPS PUB 46-3. 1999.

[8] Li H., Friggstad Z. An Efficient Architecture for the AES Mix Columns Operation. Department of Mathematics and Computer Science University of Lethbridge. 2010.

[9] Яблонский С. В., Гаврилов Г. П., Кудрявцев В. Б. Функции алгебры логики и классы Поста. Москва: Наука. 1966.

[10] Гашков С. Б., Гринчук М. И., Сергеев И. С. О построении схем сумматоров малой глубины. Дискретн. анализ и исслед. опер., сер. 1. 2007. № 14:1. С. 19–25.