УДК 003.26.09 DOI: 10.18698/2541-8009-2022-04-785

СРАВНИТЕЛЬНЫЙ АНАЛИЗ РОССИЙСКОГО СТАНДАРТА ШИФРОВАНИЯ ПО ГОСТ Р 34.12–2015 И АМЕРИКАНСКОГО СТАНДАРТА ШИФРОВАНИЯ AES

М.А. Соболев smax2011.98@mail.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Рассмотрен действующий российский алгоритм шифрования «Кузнечик» по ГОСТ Р 34.12–2015 «Информационные технологии. Криптографическая защита информации. Блочные шифры» и режимы работы данного алгоритма. Выполнено сравнение по быстродействию и криптостойкости алгоритмов шифрования «Кузнечик» по ГОСТ Р 34.12–2015 и американского криптографического стандарта AES. В качестве основного критерия сравнения использовали криптографическую стойкость алгоритма. В результате изучения российского и американского алгоритмов шифрования было установлено, что можно использовать оба стандарта, потому что несущие максимальную угрозу атаки против этих алгоритмов практически нереализуемы.

Ключевые слова

Шифр, шифрование, криптостойкость, раунды, гаммирование, алгоритм «Кузнечик», SP-сеть, стандарт AES

Поступила в редакцию 15.04.2022

Введение. Шифрование представляет собой преобразование информации в целях сокрытия ее от неавторизованных пользователей с одновременным предоставлением авторизованным пользователям доступа к ней. Шифрование необходимо для соблюдения конфиденциальности передаваемой информации. Важная особенность любого алгоритма шифрования — использование ключа, который утверждает выбор конкретного преобразования из совокупности возможных для данного алгоритма.

Пользователи являются авторизованными, если они обладают определенным аутентичным (подлинным) ключом. Задача шифрования — обеспечить доступ к информации только для авторизованных пользователей и сделать практически невозможным доступ для всех остальных.

Шифрование применяется в таких сферах как телевидение, Интернет, облачные технологии, сетевые технологии, банковская деятельность, радиосвязь и прочие коммуникации.

Шифр «Кузнечик» по ГОСТ Р 34.12–2015. ***Описание шифра.*** В российском шифре ГОСТ Р 34.12–2015 «Кузнечик» применяется SP-сеть (Substitution-Permutation network, подстановочно-перестановочная сеть) — разновидность

блочного шифра, предложенная в 1971 г. Хорстом Фейстелем. В простейшем варианте SP-сеть представляет собой «сэндвич» из слоев двух типов, используемых многократно по очереди. Слой первого типа — P-слой, состоящий из P-блока большой разрядности, за ним идет слой второго типа — S-слой, представляющий собой большое количество S-блоков малой разрядности, потом опять P-слой и т. д.

S-блок — блок подстановок, блок нелинейных замен. На вход данного блока поступает один поток бит (блок данных), а на выходе получается другой. Особенностью блока подстановок является то, что если на вход подается поток бит A, то на выходе получится поток бит B, а если на выход подать поток бит B, то на входе будет поток бит A.

P-блок — блок перестановок, блок линейных замен. Представляет собой частный случай S-блока. На выходе этого блока поток будет содержать те же биты, что и поток, поступивший на вход, но в новом потоке положение битов изменено по сравнению с входным потоком.

Шифр «Кузнечик» имеет 128-битный размер входного блока данных, 256-битный ключ и выполняет 10 раундов шифрования. В последнем раунде шифрования выполняется только одна операция — наложение раундового ключа.

В линейном преобразовании алгоритма «Кузнечик» используются операции с байтами, которые переставляются как элементы поля Галуа: $F = GF(2^8)$. В шифре «Кузнечик» преобразование осуществляется с помощью неприводимого полинома:

$$m(x) = x^8 + x^7 + x^6 + x + 1.$$

Байты-константы задаются целыми числами, двоичные разряды которых служат соответствующими коэффициентами полинома.

Сначала выполняется операция наложения раундового ключа с помощью операции побитового *XOR*: $X[k](a) = k \oplus a$.

На следующем этапе входной 128-битовый блок a алгоритма шифрования представляется в виде 16-байтной последовательности, байты которой нумеруются справа налево, начиная с нулевого байта:

$$a = a_{15} \,||\, a_{14} \,||\, \ldots \,||\, a_1 \,||\, a_0,$$

где знаком $||$ обозначена операция конкатенации строк.

Нелинейная подстановка применяется к каждому байту и задается следующим массивом:

$$\pi = (252,238,221,17,207,110,49,22,251,196,250,218,35,197,4,77,233,119,240,219,147,$$

$$46,153,186,23,54,241,187,20,205,95,193,249,24,101,90,226,92,239,33,129,28,60,66,$$

$$139,1,142,79,5,132,2,174,227,106,143,160,6,11,237,152,127,212,211,31,235,52,44,81,$$
$$234,200,72,171,242,42,104,162,253,58,206,204,181,112,14,86,8,12,118,18,191,114,19,$$
$$71,156,183,93,135,21,161,150,41,16,123,154,199,243,145,120,111,157,158,178,177,50,$$
$$117,25,61,255,53,138,126,109,84,198,128,195,189,13,87,223,245,36,169,62,168,67,201,$$
$$215,121,214,246,124,34,185,3,224,15,236,222,122,148,176,188,220,232,40,80,78,51,10,$$
$$74,167,151,96,115,30,0,98,68,26,184,56,130,100,159,38,65,173,69,70,146,39,94,85,47,$$
$$140,163,165,125,105,213,149,59,7,88,179,64,134,172,29,247,48,55,107,228,136,217,$$
$$231,137,225,27,131,73,76,63,248,254,141,83,170,144,202,216,133,97,32,113,103,164,$$
$$45,43,9,91,203,155,37,208,190,229,108,82,89,166,116,210,230,244,180,192,209,102,$$
$$175,194,57,75,99,182).$$

Каждый байт a_i принимает значение в диапазоне от 0 до 255 и массив π имеет 256 элементов, поэтому при $a_i = 0$ значение a_i будет заменено значением $\pi(0) = 252$, а при $a_i = 1$ — значением $\pi(1) = 238$ и т. д. В целом для выходного блока нелинейная подстановка определяется следующей формулой:

$$\pi(a) = \pi(a_{15} \| \ldots \| a_0) = \pi(a_{15}) \| \ldots \| \pi(a_0).$$

Линейное перемешивание *L* шифра «Кузнечик» может быть описано с помощью алгоритма линейного регистра сдвига *R*:

1) байты a_i представляют в виде полиномов из поля Галуа $GF(2^8)$;

2) в поле *F* вычисляют новый байт:

$$\gamma(a_{15},\ldots\ldots,a_0) = 148a_{15} + 32a_{14} + 133a_{13} + 16a_{12} + 194a_{11} + 192a_{10} + a_9 + 251a_8 +$$
$$+ a_7 + 192a_6 + 194a_5 + +16a_4 + 133a_3 + 32a_2 + 148a_1 + a_0;$$

3) записывают выражение, осуществляющее сдвиг вправо строки байтов:

$$R(a_{15},\ldots,a_0) = \gamma(a_{15},\ldots,a_0) \| a_{15} \| a_{14} \| \ldots \| a_1;$$

4) шаги 1–3 повторяют 16 раз; при этом получаемая строка $L(a) = R^{16}(a)$ не содержит ни одного не преобразованного символа.

В результате выполнения действий 1–4 шифрование 128-битного входного блока будет описываться следующей формулой:

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

где K_i — ключи раундов, записанные подряд преобразования; $LSX[K_i]$ — последовательное применение преобразований справа налево, т. е. сначала осуществляется прибавление ключа, затем нелинейная подстановка *S*, а после происходит линейное перемешивание *L*.

Режимы работы шифра «Кузнечик» по ГОСТ 34.10–2015. В ГОСТ Р 34.13–2015 определены шесть режимов работы алгоритмов блочного шифрования с произвольной длиной входного блока *n*.

1. *Режим простой замены* полностью аналогичен одноименному режиму ГОСТ 28147–89. Все блоки шифруются и расшифровываются независимо друг от друга.

2. В *режиме простой замены с зацеплением* применяется двоичный регистр сдвига *R* длиной *m*, где $m = nz$, а $z > 1$, где *z* — целое число. Начальным значением регистра служит значение синхропосылки IV. Очередной блок зашифрованного текста создается с помощью зашифрования результата операции побитового исключающего или (XOR) над значением очередного блока открытого текста *M* со значением *n* старших разрядов регистра сдвига. Затем происходит сдвиг влево на один блок (*n* бит) регистра *R*. В младшие разряды осуществляется запись значения блока шифротекста (рис. 1) [1].

При расшифровании блоки шифротекста *C* также заносятся в младшие биты регистра сдвига, но базовый алгоритм работает в режиме расшифрования, а затем происходит сложение по модулю полученного результата с *n* старшими разрядами регистра *R* (рис. 2) [1].

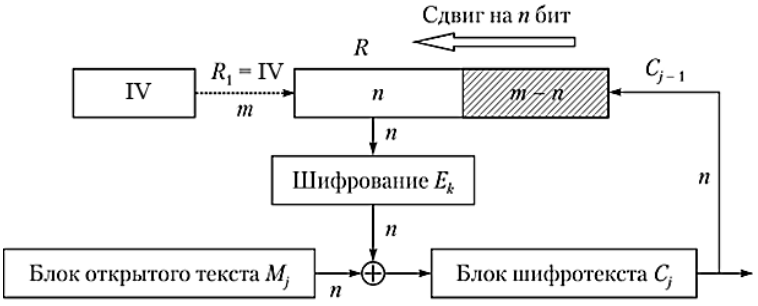

Рис. 1. Шифрование в режиме простой замены с зацеплением

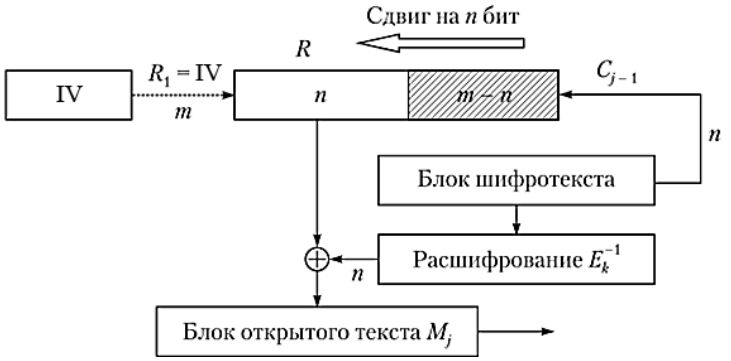

Рис. 2. Расшифровывание в режиме простой замены с зацеплением [1]

3. Существует несколько режимов, в которых применяется *гаммирование*[1]: режим гаммирования, режим гаммирования с обратной связью по выходу и режим гаммирования с обратной связью по шифротексту. Режим гаммирования с обратной связью по шифротексту позволяет шифровать данные блоками с произвольной длиной $s < m$. В перечисленных выше режимах происходит наложение гаммы шифра, вырабатываемой блоками длиной *s*, на открытый текст. Разница между этими режимами заключается только в процедуре, формирующей гамму.

При расшифровании необходимо выработать такую же гамму и наложить ее на шифротекст, поэтому базовый алгоритм здесь используется тот же, что и в режиме шифрования.

А. В *режиме гаммирования* гамму получают с помощью зашифрования последовательности значений счетчика. Этот режим аналогичен режиму CTR (Counter Mode — режим счетчика) западных блочных шифров (рис. 3) [1].

Для зашифрования (расшифрования) любого отдельного открытого текста с использованием ключа в данном режиме используется значение уникальной синхропосылки IV длина, которой равна половине длины входного блока $n/2$.

В режиме гаммирования, когда происходит шифрование нескольких сообщений с использованием одного ключа, все применяемые синхропосылки должны являться уникальными.

Начальное значение счетчика получают в результате дополнения справа нулями значения IV до полной длины блока n $\mathrm{CTR}_0 = \mathrm{IV} \,||\, 0^{n/2}$. Последующие значения счетчика вычисляют как результат сложения целочисленного значения счетчика по модулю 2^n с единицей:

$$\mathrm{CTR}_{j+1} = (\mathrm{CTR}_j + 1) \bmod 2^n.$$

Б. *Режим гаммирования с обратной связью по выходу* показан на рис. 3. В данном режиме применяется двоичный регистр сдвига *R*, имеющий длину *m*, $m = nz$, где $z > 1$ — целое число. Начальным значением регистра *R* является значение синхропосылки IV длиной *m*. Синхропосылка должна быть непредсказуемой, т. е. случайной или псевдослучайной, либо уникальной [1].

Старшие *n* бит регистра сдвига шифруются, затем *s* старших битов результата шифрования *Y* применяются в качестве гаммы шифра O_j. На следующем шаге происходит сдвиг влево на *n* бит регистра *R*, а в область младших битов записывается значение Y_j.

---

[1] *Гаммирование* — это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы. *Криптографическая гамма* — последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных.

В. *Режим гаммирования с обратной связью по зашифрованному тексту* показан на рис. 4. В данном режиме применяется двоичный регистр сдвига R длиной m, $m > n$. Начальным значением регистра служит значение синхропосылки IV.

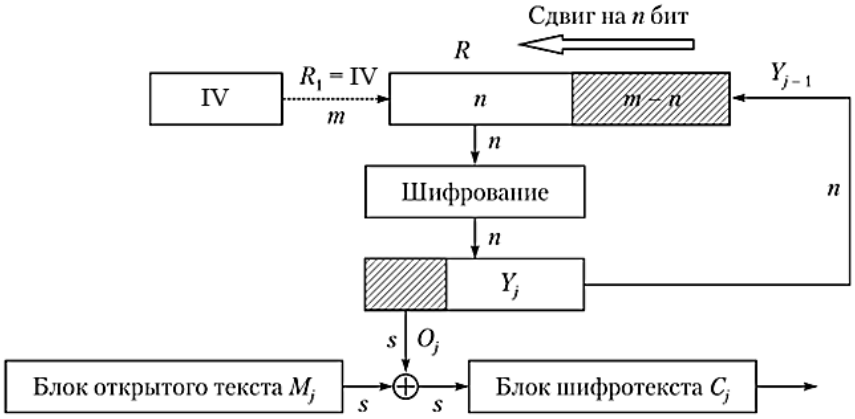

Рис. 3. Режим гаммирования с обратной связью по выходу [1]

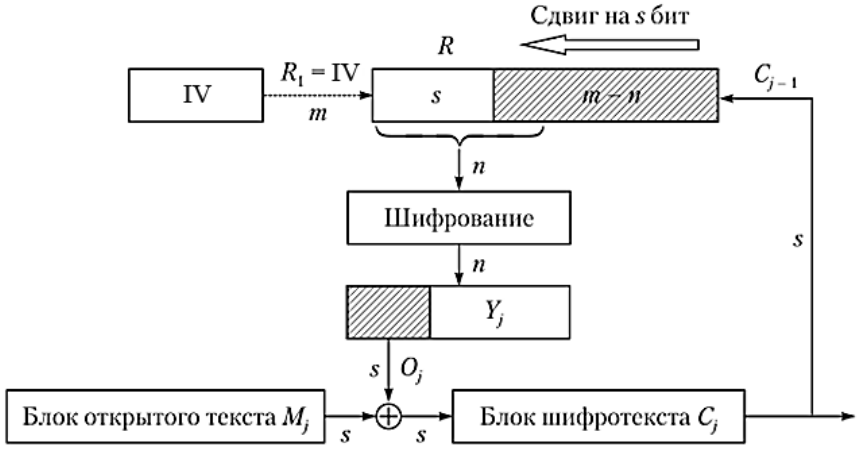

Рис. 4. Режим гаммирования с обратной связью по шифротексту (ГОСТ Р 34.13–2015) [1]

Также у шифра «Кузнечик» существует еще один режим — *режим выработки имитовставки*.

Процедура вычисления значения имитовставки похожа на процедуру зашифрования в режиме простой замены с зацеплением, показанную на рис. 2, при $m = n$ и инициализации начального заполнения регистра сдвига R значением 0^n. В режиме имитовставки на вход алгоритма шифрования подается результат покомпонентного сложения очередного блока открытого текста[2] и ре-

[2] Данных, подлежащих шифрованию.

зультата зашифрования на предыдущем шаге. Основное отличие от зашифрования в режиме простой замены с зацеплением заключается в процедуре обработки последнего блока данных: на вход алгоритма блочного шифрования подается результат покомпонентного сложения последнего блока данных, результата зашифрования на предыдущем шаге и одного из производных ключей[3]. Выбор конкретного производного ключа зависит от того, является ли последний блок исходного сообщения полным или нет. Значением имитовставки является результат применения процедуры усечения к выходу алгоритма шифрования при обработке последнего блока.

Сравнение шифра AES и шифра «Кузнечик» по ГОСТ Р 34.11–2015. В алгоритме AES могут использоваться ключи следующих размеров: 128, 192 и 256 бит. Шифр AES имеет 10 раундов для 128-битных ключей, 12 раундов для 192-битных ключей и 14 раундов для 256-битных ключей.

Алгоритм «Кузнечик» имеет длину ключа 256 бит, длину блока обрабатываемых данных 128 бит и 9 раундов.

Принципиальное различие шифров AES и «Кузнечик» заключается в алгоритме развертки ключа. Алгоритм развертки AES представляет простую процедуру, в которой раундовые ключи формируются как конкатенации 32-битных «слов», вырабатываемых с помощью рекуррентной процедуры. В алгоритме «Кузнечик» для развертки ключа последовательно применяется преобразование *F*, которое представляет собой сеть Фейстеля. Каждое применение преобразования *F* позволяет выработать сразу пару раундовых ключей.

Алгоритмы зашифрования шифра AES и российского шифра «Кузнечик» могут быть представлены в виде последовательного применения *r*-раундовых преобразований $R(k_i, a)$, где $i = 1, \ldots, r$, где $r = 14$ для шифра AES и $r = 9$ для шифра «Кузнечик». Преобразование имеет вид

$$R(k_i, a) = L\Big(S\big(X(k_i, a)\big)\Big),$$

где X — преобразование наложения ключа k_i; S — нелинейная перестановка на множестве $\mathbb{F}_{128}^2$, определенная следующим выражением $S(a) = \pi(a_1) \| \ldots \| \pi(a_{16})$. Для каждого шифра используется собственная нелинейная перестановка $\pi : \mathbb{F}_8^2 \to \mathbb{F}_8^2$ [2].

Преобразование L для обоих шифров может быть представлено в виде произведения вектора $a \in \mathbb{F}_{128}^2$ и квадратной матрицы $\mathcal{L}$.

Для шифра AES матрица $\mathcal{L}$ выглядит следующим образом [2]:

---

[3] Ключей, полученных из исходного ключа при проведении процедуры шифрования.

$$\begin{pmatrix} \alpha_0 & 0 & 0 & 0 & 0 & \alpha_3 & 0 & 0 & 0 & 0 & \alpha_2 & 0 & 0 & 0 & 0 & \alpha_1 \\ 0 & \alpha_0 & 0 & 0 & 0 & 0 & \alpha_3 & 0 & 0 & 0 & 0 & \alpha_2 & \alpha_1 & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & 0 & \alpha_3 & 0 & 0 & 0 & 0 & \alpha_2 & \alpha_1 & 0 & 0 & 0 & 0 & \alpha_0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_3 & \alpha_2 & 0 & 0 & 0 & 0 & \alpha_1 & 0 & 0 & 0 & 0 & \alpha_0 & 0 \end{pmatrix}$$

Для шифра «Кузнечик» матрица $\mathcal{L}$ выглядит так, как показано ниже [2].

$$\begin{pmatrix} 0x01 & 0x94 & 0x20 & 0x85 & 0x10 & 0xC2 & 0xC0 & 0x01 & 0xFB & 0x01 & 0xC0 & 0xC2 & 0x10 & 0x85 & 0x20 & 0x94 \\ 0x94 & 0xA5 & 0x3C & 0x44 & 0xD1 & 0x8D & 0xB4 & 0x54 & 0xDE & 0x6F & 0x77 & 0x5D & 0x96 & 0x74 & 0x2D & 0x84 \\ 0x84 & 0x64 & 0x48 & 0xDF & 0xD3 & 0x31 & 0xA6 & 0x30 & 0xE0 & 0x5A & 0x44 & 0x97 & 0xCA & 0x75 & 0x99 & 0xDD \\ 0xDD & 0x0D & 0xF8 & 0x52 & 0x91 & 0x64 & 0xFF & 0x7B & 0xAF & 0x3D & 0x94 & 0xF3 & 0xD9 & 0xD0 & 0xE9 & 0x10 \\ 0x10 & 0x89 & 0x48 & 0x7F & 0x91 & 0xEC & 0x39 & 0xEF & 0x10 & 0xBF & 0x60 & 0xE9 & 0x30 & 0x5E & 0x95 & 0xBD \\ 0xBD & 0xA2 & 0x48 & 0xC6 & 0xFE & 0xEB & 0x2F & 0x84 & 0xC9 & 0xAD & 0x7C & 0x1A & 0x68 & 0xBE & 0x9F & 0x27 \\ 0x27 & 0x7F & 0xC8 & 0x98 & 0xF3 & 0x0F & 0x54 & 0x08 & 0xF6 & 0xEE & 0x12 & 0x8D & 0x2F & 0xB8 & 0xD4 & 0x5D \\ 0x5D & 0x4B & 0x8E & 0x60 & 0x01 & 0x2A & 0x6C & 0x09 & 0x49 & 0xAB & 0x8D & 0xCB & 0x14 & 0x87 & 0x49 & 0xB8 \\ 0xB8 & 0x6E & 0x2A & 0xD4 & 0xB1 & 0x37 & 0xAF & 0xD4 & 0xBE & 0xF1 & 0x2E & 0xB8 & 0x1A & 0x4E & 0xE6 & 0x7A \\ 0x7A & 0x16 & 0xF5 & 0x52 & 0x78 & 0x99 & 0xEB & 0xD5 & 0xE7 & 0xC4 & 0x2D & 0x06 & 0x17 & 0x62 & 0xD5 & 0x48 \\ 0x48 & 0xC3 & 0x02 & 0x0E & 0x58 & 0x90 & 0xE1 & 0xA3 & 0x6E & 0xAF & 0xBC & 0xC5 & 0x0C & 0xEC & 0x76 & 0x6C \\ 0x6C & 0x4C & 0xDD & 0x65 & 0x01 & 0xC4 & 0xD4 & 0x8D & 0xA4 & 0x02 & 0xEB & 0x20 & 0xCA & 0x6B & 0xF2 & 0x72 \\ 0x72 & 0xE8 & 0x14 & 0x07 & 0x49 & 0xF6 & 0xD7 & 0xA6 & 0x6A & 0xD6 & 0x11 & 0x1C & 0x0C & 0x10 & 0x33 & 0x76 \\ 0x76 & 0xE3 & 0x30 & 0x9F & 0x6B & 0x30 & 0x63 & 0xA1 & 0x2B & 0x1C & 0x43 & 0x68 & 0x70 & 0x87 & 0xC8 & 0xA2 \\ 0xA2 & 0xD0 & 0x44 & 0x86 & 0x2D & 0xB8 & 0x64 & 0xC1 & 0x9C & 0x89 & 0x48 & 0x90 & 0xDA & 0xC6 & 0x20 & 0x6E \\ 0x6E & 0x4D & 0x8E & 0xEA & 0xA9 & 0xF6 & 0xBF & 0x0A & 0xF3 & 0xF2 & 0x8E & 0x93 & 0xBF & 0x74 & 0x98 & 0xCF \end{pmatrix}$$

Академический руководитель образовательной программы «Компьютерная безопасность» НИУ «Высшая школа экономики» А.Б. Лось делает вывод, что алгоритм «Кузнечик» обладает ***плотной*** матрицей $\mathcal{L}$, т. е. матрицей, не содержащей большого числа нулей. Матрица алгоритма AES является ***разреженной***, т. е. в основном состоящей из нулей. А.Б. Лось делает вывод, что различие в матрицах линейного преобразования американского шифра и российского шифра привело к тому, что число раундов алгоритма «Кузнечик» существенно меньше, чем в алгоритме AES.

Опишем некоторые атаки, которые были совершены против российского шифра «Кузнечик» и американского шифра AES.

На конференции “CRYPTO 2015” криптографы Алекс Бирюков, Лео Перрин и Алексей Удовенко представили доклад, в котором утверждается, что значения S-блока российского шифра «Кузнечик» и российской хеш-функции «Стрибог» не являются псевдослучайными числами, а сгенерированы на основе скрытого алгоритма, который им удалось восстановить методами обратного проектирования [3].

Канадские специалисты в области криптографии Рихам Аль-Тави (Riham AlTawy) и Амр М. Юссеф (Amr M. Youssef) описали атаку «встречи на середине» для пяти раундов шифра «Кузнечик», имеющую вычислительную сложность 2^{140} операций и требующую 2^{153} памяти и 2^{113} данных [4].

На процессоре Intel i7-5820 скорость работы шифра «Кузнечик» равна 160 Мб/с [5].

Первые атаки с восстановлением ключа на полный AES были проведены европейскими исследователями Андреем Богдановым, Дмитрием Ховратовичем и Кристианом Рехбергером. Результаты их работ были опубликованы в 2011 г. [6]. Эта атака представляет собой атаку типа «встреча на середине» и протекает примерно в 4 раза быстрее, чем полный перебор. Для восстановления ключа AES-128 требуется $2^{126,2}$ операций. Для AES-192 и AES-256 необходимо $2^{190,2}$ и $2^{254,6}$ операций соответственно. Этот результат был дополнительно улучшен до $2^{126,0}$ операций для AES-128, $2^{189,9}$ для AES-192 и $2^{254,3}$ для AES-256 [7]. В настоящее время, полученные при проведении атаки, на полный AES результаты по количеству вычислительных операций являются лучшими (для их достижения требуется наименьшее число операций процессора) при атаке с восстановлением ключа против AES. Это очень небольшой выигрыш, поскольку для перебора 126-битного ключа (вместо 128-битного) на существующем (и перспективном в обозримом будущем) оборудовании по-прежнему будут требоваться миллиарды лет. Также авторы [7], описавшие наилучшую атаку против шифра AES, рассчитывают, что наилучшая атака по их методике на AES со 128-битным ключом требует хранения 2^{88} бит данных. Это составляет около 38 трлн Тб, что больше, чем все данные, хранящиеся на всех компьютерах на планете в 2016 г. Таким образом, нет никаких практических последствий для безопасности AES, т е. взломать его за допустимое время невозможно [8]. Сложность пространства позже была улучшена до 2^{56} бит [7], что составляет 9007 Тб.

В настоящее время не существует известной практической атаки, которая позволила бы кому-то, не знающему ключа, прочитать данные, зашифрованные с помощью алгоритма AES при правильной реализации данного алгоритма.

Для реализации шифрования AES на процессоре Pentium Pro требуется 18 тактов на байт [9], что эквивалентно пропускной способности около 11 МБ/с для процессора с тактовой частотой 200 МГц.

На процессорах Intel Core i7 6900K скорость шифрования AES составляет 11,8 Гб/с, а для процессора AMD Ryzen скорость работы шифра AES составляет 10,7 Гб/с [10].

Скорость работы шифра AES выше, чем у шифра «Кузнечик», но данный параметр зависит от процессора, на котором был запущен алгоритм шифрования, и поэтому не является ключевым. Другим более важным параметром служит стойкость к криптографическим атакам. Атака «встреча на середине» требует большего количества итераций для шифра AES, чем для шифра «Кузнечик». Однако аналогично шифру AES-128 не существует никаких практических последствий атаки против шифра «Кузнечик», поэтому можно применять на практике как российский стандарт шифрования, так и американский.

Выводы. В рамках статьи были рассмотрен российский шифров «Кузнечик» и описаны режимы работы данного криптографического алгоритма. Проведено сравнение российского стандарта шифрования и американского стандарта шифрования по быстродействию и устойчивости к криптографическим атакам против данных шифров. Сделан вывод, что на практике можно использовать оба стандарта шифрования, поскольку они удовлетворяют современным требованиям безопасности.

Литература

[1] Бабенко Л.К., Ищукова Е.А. Криптографическая защита информации: симметричное шифрование. М., Юрайт, 2019.

[2] Лось А.Б., Нестеренко А.Ю., Рожков М.И. Криптографические методы защиты информации для изучающих компьютерную безопасность. М., Юрайт, 2019.

[3] Biryukov A., Perrin L., Udovenko A. Reverse-engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1. In: *Advances in cryptology – EUROCRYPT 2016*. Springer, 2016, pp. 372–402. DOI: https://doi.org/10.1007/978-3-662-49890-3_15

[4] Altawy R., Youssef A.M. A meet in the middle attack on reduced round Kuznyechik. *IEICE Trans. Fundam. Electron.*, 2015, vol. E98.A, no. 10, pp. 2194–2198. DOI: https://doi.org/10.1587/transfun.E98.A.2194

[5] Овчинников А.И., Мелешин А.Е., Истомин А.А. Исследование различных характеристик шифра «Кузнечик» на российских процессорах и платформах IoT. URL: https://www.ruscrypto.ru/resource/archive/rc2018/files/10_Ovchinnikov.pdf (дата обращения: 22.01.2021).

[6] Bogdanov A., Khovratovich D., Rechberger C. Biclique cryptanalysis of the full AES. In: *ASIACRYPT 2011*. Springer, 2011, 344–375.
DOI: https://doi.org/10.1007/978-3-642-25385-0_19

[7] *Tao B., Wu H. Improving the biclique cryptanalysis of AES. In: ACISP 2015. Springer, 2015, pp. 39–56. DOI:* https://doi.org/10.1007/978-3-319-19962-7_3

[8] *Goldberg J. AES encryption isn't cracked. blog.1password.com: веб-сайт. URL: https://blog.1password.com/aes-encryption-isnt-cracked/ (дата обращения 24.01.2022).*

[9] *Schneier B., Kelsey J., Whiting D. Et al. Performance comparisons of the AES submissions. URL: https://www.schneier.com/wp-content/uploads/2016/02/paper-aes-performance.pdf (дата обращения 24.01.2022).*

[10] *AMD Ryzen 7 1700X review. vortez.ne: веб-сайт.*
URL: https://www.vortez.net/articles_pages/amd_ryzen_7_1700x_review,7.html (дата обращения 24.01.2022).

Соболев Максим Алексеевич — магистрант кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Сидякин Иван Михайлович, кандидат технических наук, доцент кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:
Соболев М.А. Сравнительный анализ российского стандарта шифрования по ГОСТ Р 34.12–2015 и американского стандарта шифрования AES. *Политехнический молодежный журнал*, 2022, № 04(69). http://dx.doi.org/10.18698/2541-8009-2022-04-785

COMPARATIVE ANALYSIS OF RUSSIAN GOST R 34.12-2015 ENCRYPTION STANDARD AND AMERICAN ENCRYPTION STANDARD AES

M.A. Sobolev smax2011.98@mail.ru

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The current Russian encryption algorithm "Grasshopper" according to GOST R 34.12-2015 "Information technologies. Cryptographic protection of information. Block ciphers" and operation modes of this algorithm are considered. The comparison of the speed and cryptographic strength of the encryption algorithm "Grasshopper" according to GOST R 34.12-2015 and the American cryptographic standard AES was performed. Cryptographic strength of algorithm was used as the main criterion of comparison. As a result of studying Russian and American encryption algorithms, it was found that both standards can be used because maximum-threat attacks against these algorithms are practically unrealizable.

Keywords

Cipher, encryption, cryptographic strength, rounds, gamming, "Grasshopper" algorithm, SP-net, AES standard

Received 15.04.2022

References

[1] Babenko L.K., Ishchukova E.A. Kriptograficheskaya zashchita informatsii: simmetrichnoe shifrovanie [Cryptographic information protection: symmetric encryption]. Moscow, Yurayt Publ., 2019 (in Russ.).

[2] Los' A.B., Nesterenko A.Yu., Rozhkov M.I. Kriptograficheskie metody zashchity informatsii dlya izuchayushchikh komp'yuternuyu bezopasnost' [Cryptographic methods of information protection for those who study computer security]. Moscow, Yurayt Publ., 2019 (in Russ.).

[3] Biryukov A., Perrin L., Udovenko A. Reverse-engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1. In: Advances in cryptology – EUROCRYPT 2016. Springer, 2016, pp. 372–402. DOI: https://doi.org/10.1007/978-3-662-49890-3_15

[4] Altawy R., Youssef A.M. A meet in the middle attack on reduced round Kuznyechik. IEICE Trans. Fundam. Electron., 2015, vol. E98.A, no. 10, pp. 2194–2198. DOI: https://doi.org/10.1587/transfun.E98.A.2194

[5] Ovchinnikov A.I., Meleshin A.E., Istomin A.A. Issledovanie razlichnykh kharakteristik shifra "Kuznechik" na rossiyskikh protsessorakh i platformakh IoT [Study on different characteristics of Kuznyechik cipher on Russian processors and platforms] (in Russ.). URL: https://www.ruscrypto.ru/resource/archive/rc2018/files/10_Ovchinnikov.pdf (accessed: 22.01.2021).

[6] Bogdanov A., Khovratovich D., Rechberger C. Biclique cryptanalysis of the full AES. In: ASIACRYPT 2011. Springer, 2011, 344–375. DOI: https://doi.org/10.1007/978-3-642-25385-0_19

[7] Tao B., Wu H. Improving the biclique cryptanalysis of AES. In: ACISP 2015. Springer, 2015, pp. 39–56. DOI: https://doi.org/10.1007/978-3-319-19962-7_3
[8] Goldberg J. AES encryption isn't cracked. blog.1password.com: website. URL: https://blog.1password.com/aes-encryption-isnt-cracked/ (accessed 24.01.2022).
[9] Schneier B., Kelsey J., Whiting D. Et al. Performance comparisons of the AES submissions. URL: https://www.schneier.com/wp-content/uploads/2016/02/paper-aes-performance.pdf (accessed 24.01.2022).
[10] AMD Ryzen 7 1700X review. vortez.ne: website. URL: https://www.vortez.net/articles_pages/amd_ryzen_7_1700x_review,7.html (accessed 24.01.2022).

Sobolev M.A. — Student, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Sidyakin I.M., Cand. Sc. (Eng.), Assoc. Professor, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:
Sobolev M.A. Comparative analysis of russian gost r 34.12-2015 encryption standard and american encryption standard aes. *Politekhnicheskiy molodezhnyy zhurnal* [Politechnical student journal], 2022, no. 04(69). http://dx.doi.org/10.18698/2541-8009-2022-04-785.html (in Russ.).