Моделирование атаки внесением ошибок на шифр ГОСТ 34.12–2015 с длиной блока 128 бит

В. А. Суворова
(Пермский государственный национальный исследовательский университет)

В статье описывается атака на шифр ГОСТ 34.12–2015 с длиной блока 128 бит, реализуемая путем внесения ошибок в процесс зашифрования. Показано, что с помощью анализа внесенных ошибок при известных блоках замены можно вычислить значение ключа в среднем за 2^{13} инъекций неисправностей, используя около 128 Кб открытого текста.

Ключевые слова: криптоанализ, ГОСТ 34.12–2015, атака внесением ошибки, атака на блочный шифр.

Введение

Блочный шифр ГОСТ 34.12–2015, называемый также «Кузнечик» [1], был утвержден в качестве нового стандарта шифрования в Российской Федерации с 2016 года. Предыдущий стандарт ГОСТ 28147–89 [2], используемый с 1990, многократно подвергался критике со стороны зарубежных ученых [3, 4], однако реализуемые на практике атаки так и не были представлены. ГОСТ 28147–89 оперирует блоками данных длиной 64 бита, что не всегда соответствует современным требованиям, поэтому возникла необходимость введения нового стандарта с большей длиной обрабатываемого блока, которым стал шифр «Кузнечик».

Криптографические свойства нового стандарта шифрования уже подтверждены рядом научно-исследовательских работ [5, 6]. В частности, доказано, что алгоритм не подвержен большей части типичных атак на блочные шифры. Тем не менее, при определенных условиях имеется возможность реализации атаки, осуществимой с помощью аппаратных средств, например, атаки с помощью внесения ошибок в процесс зашифрования (расшифрования).

Идея атаки

Суть атаки заключается в анализе внесенных в процесс шифрования ошибок. В данном контексте под внесением ошибки будем понимать сброс в ноль некоторых бит внутреннего состояния некоторого регистра шифрующего устройства путем физического вмешательства. Данная атака опирается на результаты исследований, описанные авторами Riham AlTawy, Onur Duman, и Amr M. Youssef в статье [7].

Для реализации атаки имеем следующие предположения:

- атакующий имеет доступ к регистрам устройства шифрования: он не знает их значения, но может обнулить состояние конкретного регистра в конкретный момент времени;
- атакующий может неограниченное число раз подавать на вход шифрующему устройству различные открытые тексты и подучать соответствующие зашифрованные сообщения;
- устройство шифрования в своей реализации использует известные блоки перестановок, предложенные в стандарте;
- внутри устройства шифрования хранится ключ K, который используется для шифрования, и атакующий стремится узнать значение K.

Далее необходимо обратиться к стандарту шифрования ГОСТ 34.12–2015 [1] с длиной блока $n = 128$ бит, ведь дальнейшие действия зависят от спецификации алгоритма. Алгоритм шифрования известен, следовательно, известны действия, выполняемые внутри шифрующего устройства (см. рис. 1).

Как видно из рисунка 1, сразу после попадания в шифрующее устройство текущий блок открытого текста складывается по модулю 2 с итерационным ключом K_1 (преобразование X), а затем попадает на вход блоку перестановок, который является известным (нелинейное биективное преобразование S), затем текущий блок обрабатывается с помощью линейного преобразования L, и эти действия повторяются в цикле 9 раз с ключами $K_1, \ldots, K_9$. После чего результат складывается по модулю 2 с итерационным ключом K_{10} и подается на выход как полученный шифртекст.

Важно также отметить, что для нахождения мастер ключа K достаточно определить лишь значения итерационных ключей K_1, K_2, так как $K = K_1 \parallel K_2$.

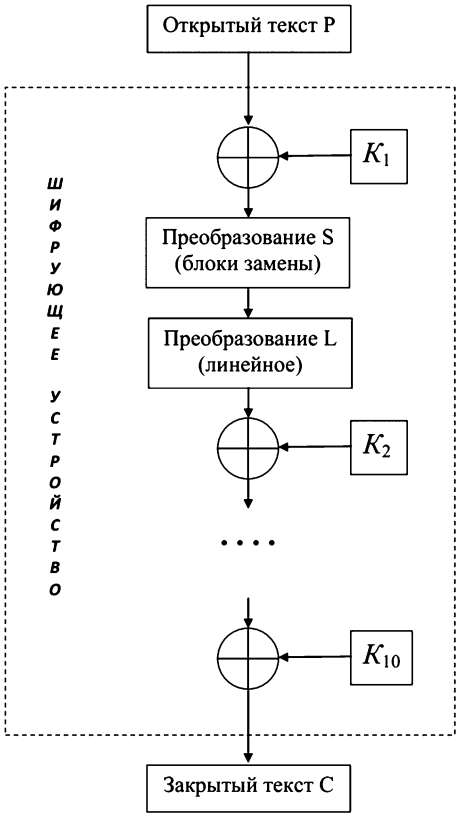

Рис. 1. Подробное изображение процедуры шифрования.

Для нахождения i-ого байта ключа K_1, можно вносить ошибку в i-ый байт обрабатываемого блока, после его прохождения через S-блок, перебирая различные входные сообщения до тех пор, пока не получим ошибку, не повлекшую изменений. Ошибкой, не повлекшей изменений (ineffective fault), будем называть такую ошибку, в результате внесения которой в процесс шифрования, полученный зашифрованный текст не отличается от исходного, то есть, $C = C'$.

Если возникает ошибка, не повлекшая изменений, при обнулении некоторого байта обрабатываемого сообщения после его прохождения через S-блоки, значит, соответствующий байт до попадания в S-блок был равен $S^{-1}(0)$.

Следовательно i-ый байт ключа K_1 тогда может быть найден по формуле (1).

$$K_1[i] = P[i] \oplus S^{-1}(0), \tag{1}$$

где $P[i]$ — i-ый байт открытого текста, $S^{-1}(0)$ — обратная перестановка от нулевого байта (прообраз нуля в блоке замены).

Таким образом, если выполнить данную атаку для каждого из 16-ти байт итерационного ключа K_1, можно полностью узнать его значение. Для нахождения каждого из 16-ти байт K_1 необходимо повторить в среднем 2^8 инъекций (внесений ошибок) с различными открытыми текстами, перебирая все возможные значения данного байта открытого текста. Следовательно, для нахождения K_1 необходимо $16 \times 2^8 = 2^{12}$ попыток.

После нахождения ключа K_1 нетрудно вычислить аналогичным образом все байты ключа K_2, внося ошибку в текущий обрабатываемый блок на втором раунде шифрования.

Обозначим $P_1 = LSX[K_1](P)$ — блок открытого текста, прошедший один раунд шифрования. Тогда K_2 может быть найден по формуле (2):

$$K_2[i] = P_1[i] \oplus S^{-1}(0). \tag{2}$$

Очевидно, что для нахождения K_2 также необходимо $16 \times 2^8 = 2^{12}$ попыток. Это значит, что для нахождения мастер-ключа K требуется повторить $2 \times 2^{12} = 2^{13}$ итераций внесений ошибок с различными открытыми текстами, что в 2^{243} раз меньше, чем полный перебор всех возможных значений секретного ключа (2^{256}).

Описанная общая идея атаки предполагает наличие шифрующего устройства и внесение ошибки путем физического доступа к регистрам данного шифрующего устройства. В данной работе описанная атака была смоделирована полностью программными средствами, без использования реальных физических шифрующих устройств.

Эталонная реализация алгоритма шифрования ГОСТ 34.12–2015 с длиной блока 128 бит, взятая с сайта Российского Технического комитета по стандартизации «Криптографическая защита информации» (http://tc26.ru/standard/gost/), была использована для написания программы, моделирующей данную атаку. Процедуры зашифрования и расшифрования используют известные блоки подстановок, предложенные в стандарте [1].

Тестирование программы было произведено на множестве из 10 различных ключей $K_1, \ldots, K_{10}$, сгенерированных случайным образом. Для каждого из ключей была осуществлена описанная атака, и каждый из

ключей был успешно найден за некоторое конечное число попыток. Так как ключи сгенерированы случайно и для осуществления атаки использовались открытые тексты, генерируемые случайно, то количество попыток внесения ошибки для нахождения данных ключей отличаются.

Реальные результаты количества предпринятых попыток для получения ошибки, не повлекшей изменений, для каждого байта всех тестированных ключей были сохранены. В таблице 1 представлены средние значения количества попыток для нахождения одного байта соответствующего ключа (то есть, среднее количество внесений ошибок для получения одного байта ключа).

Таблица 1. Статистические результаты нахождения ключей.

Ключ	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}
$M[K_i]$	344	239	286	278	285	205	198	259	233	228

Далее было вычислено математическое ожидание числа попыток для нахождения любого байта произвольного ключа:

$$M[K] = \frac{1}{10}\sum_{i=1}^{10} M[K_i] = 255,99 \approx 2^8. \tag{3}$$

Полученное значение говорит о том, что полученные результаты совпадают с ожидаемым, теоретическим результатом. Так как длина каждого из ключей $K_1, \ldots, K_{10}$ совпадает и равна 32 байтам, для нахождения каждого из них понадобилось в среднем $32 \times 2^8 = 2^{13}$ попыток. При этом для каждой попытки был сгенерирован новый блок открытого текста длиной 128 бит, следовательно, всего для осуществления атаки потребовалось в среднем 128×2^{13} бит = 128 Кб.

Заключение

Программная модель атаки, основанной на внесении ошибок и их анализе в процессе зашифрования данных, была успешно реализована, в ходе анализа полученных результатов выполнения атаки было показано, что с помощью внесения ошибок в процесс зашифрования при известных блоках замены можно вычислить значение ключа в среднем за 2^{13} инъекций неисправностей против 2^{256} попыток при атаке полным перебором, используя при этом около 128 Кб открытого текста.

Данная атака предполагает физический доступ нарушителя к шифрующему устройству, таким образом, акцентирует внимание на важности защиты аппаратных средств шифрования данных от проникновения и внесения изменений.

Список литературы

[1] ГОСТ Р 34.12–2015. Информационная технология. Криптографическая защита информации. Блочные шифры. Введ. 19.06.2015 г. № 749-ст. — М.: Стандартинформ, 2015.

[2] ГОСТ 28147–89. Системы обработки информации. Алгоритм криптографического преобразования. Введ. 1.07.1990. М.: ИПК изд-во стандартов, 1989.

[3] Courtois N. T. Security Evaluation of GOST 28147–89 in View of International Standardisation. [Эл. ресурс]. — URL: https://eprint.iacr.org/2011/211.pdf (дата обращ.: 05.11.2016).

[4] Zhao X., Guo S., Zhang F., Wang T., Shi Z., Gu D. Algebraic fault analysis on GOST for key recovery and reverse engineering // IEEE workshop on Fault Diagnosis and Tolerance in Cryptography. — 2014. — P. 29–39.

[5] AlTawy R., Youssef A. M. A Meet in the Middle Attack on Reduced Round Kuznyechik. [Эл. ресурс]. — URL: https://eprint.iacr.org/2015/096.pdf (дата обращ.: 06.11.2016).

[6] Biryukov A., Perrin L., Udovenko A. Reverse-Engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1 (Full Version). [Эл. ресурс]. — 2015. — URL: https://eprint.iacr.org/2016/071.pdf (дата обращения: 06.11.2016).

[7] AlTawy R., Duman O., Youssef A. M. Fault Analysis of Kuznyechik. [Эл. ресурс]. — 2015. — URL: https://eprint.iacr.org/2015/347.pdf (дата обращения: 06.11.2016).