Особенности выбора базовых операций при построении симметричных блочных шифров

Роман Елисеев[1], Роман Олейников[2]

1. Кафедра безопасности информационных технологий,
Харьковский национальный университет радиоэлектроники,
УКРАИНА, г. Харьков, пр. Науки, 14,
E-mail: eliseev.roman371@gmail.com,

2. АО «ИИТ»,
УКРАИНА, г. Харьков, ул. Бакулина, 12,
E-mail: roliynykov@gmail.com

Block cipher algorithm can be created from wide range of operations. The main limitation – it must be present on all target platforms or implementer must be able to efficiently emulate it. Despite large amount of instructions in modern CPUs only few of them used in mainstream ciphers. Its addition, rotation, XOR and array (table) indexing mainly. But many of practical ciphers combines them with additional data manipulations to achieve better performance and/or security.

Ключевые слова – блочный шифр; операция; умножение; побитовая операция; возведение в степень; дискретный логарифм.

I. Введение

На сегодняшний день блочные шифры являются одним из основных средств защиты при передаче и хранении информации. Все современные стандарты проектируются с учетом возможности работы на широком спектре оборудования, однако существенные отличия между, например, смарт-картой, RFID меткой, смартфоном и сервером приводят к тому, что спроектировать одинаково подходящий всем им алгоритм сложно и приходится иметь дело с компромиссами.

Распространяются они как на объемы обрабатываемой информации, так и на доступность элементов, на основе которых может быть построен алгоритм.

Большинство современных государственных и международных стандартов базируется на использовании блоков подстановки в качестве основной нелинейной операции. Однако все большее распространение получают алгоритмы на основе простых арифметических и логических операций процессоров.

Связано это в первую очередь с тем, что такие шифры обычно производительнее и компактнее в программной реализации, но при этом могут быть достаточно просто реализованы в аппаратном исполнении.

При этом разные платформы могут иметь различный уровень поддержки математических операций. Так, например, современные процессоры общего назначения имеют наборы векторных инструкций, позволяющих за один раз производить операции над несколькими наборами операндов, в то время как некоторые микроконтроллеры не имеют даже аппаратной поддержки умножения.

II. Обзор операций

Среди операций, не нашедших широкого применения в симметричных блочных шифрах, можно выделить умножения по модулю и побитовые вращения на переменное количество бит. Хорошим примером шифра, который использует обе операции является RC6[1], признанный одним из самых простых и красивых с математической точки зрения алгоритмов. Его предшественник, RC5[2] (Рис. 1), использует побитовые вращения на значения, зависящие от данных. К известным «пользователям» модульного умножения относится и IDEA (Рис. 2) [3].

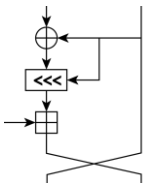

Рисунок 1. Раундовая функция RC5

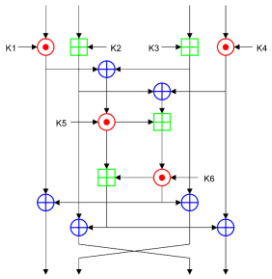

Рисунок 2. Раундовая функция IDEA

Обе эти операции сложны для классического криптоанализа и поэтому могут рассматриваться как перспективные при комбинировании с «основными», о чем говорит история IDEA [4], но в то же время они сложны и для реализации на некоторых типах устройств (некоторые виды маломощных микроконтроллеров, смарт-карты, RFID метки).

Связано это в первую очередь с тем, что некоторые узкоспециализированные архитектуры могут не включать операции умножения. Так же, в процессорах общего назначения операция умножения несколько медленнее простых арифметических операций [5].

На ряду с ней возникает проблема с атаками по времени выполнения [6], что связано в первую очередь с попытками процессора оптимизировать тяжелую операцию в среднем случае. Все эти проблемы касаются и побитовых сдвигов [7].

Другим интересным примером является SAFER [8] (Рис. 3), блоки подстановки которого основаны на возведении в степень и дискретном логарифме по модулю 257.

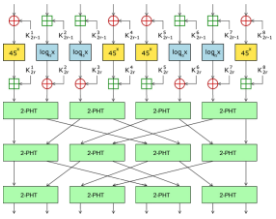

Рисунок 3. Раундовая функция SAFER

В случае же с этим шифром, проблемы возникают не на 8-битных микроконтроллерах (при условии предварительно рассчитанных таблиц), а на всех остальных платформах, так как все операции в шифре оперируют байтами и использование более широких слов не дает преимуществ, а часто еще и замедляет алгоритм в силу ручных приведений по модулю (при отсутствии 8-битных регистров в архитектуре). Кроме того, полученные таким методом блоки подстановки уступают псевдослучайным с точки зрения стойкости к ряду атак.

К малораспространенным на данный момент операциям в шифрах можно отнести и побитовые операции И и ИЛИ, а также нециклические сдвиги. Примерами шифров, использующих операцию И, являются RC2[9], SIMON[10] (Рис. 5), MISTY1[11]. Последний алгоритм использует так же и ИЛИ.

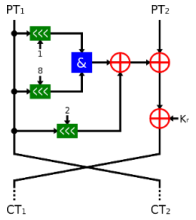

Рисунок 5. Раундовая функция SIMON

Одними из алгоритмов, которые используют побитовые нециклические сдвиги, являются SERPENT [12] и XTEA [13].

Преимуществами побитовых операций является их простота и доступность, а к недостаткам можно отнести низкий лавинный эффект [14], что приводит к необходимости использования большего количества операций по сравнению с, например, более сложным, но в то же время примерно таким же быстрым (на программных платформах), сложением по модулю 2^n.

Выводы

Использование сложных операций в блочных шифрах имеет как преимущества в виде повышения производительности на некоторых платформах или повышения стойкости к отдельным типам атак, но также и недостатки вроде появления новых векторов атак (например, атак по времени выполнения) и усложнения реализации на некоторых платформах.

Так же это усложняет криптоанализ алгоритма в силу меньшего количества уже известной информации об операциях, их взаимодействии с другими операциями.

Литература

[1] R.L. Rivest, M.J.B. Robshaw, R.Sidney, and Y.L. Yin. The RC6 Block Cipher. v1.1, August 1998.

[2] Rivest, R. L. (1994). "The RC5 Encryption Algorithm" (pdf). Proceedings of the Second International Workshop on Fast Software Encryption (FSE) 1994e: 86–96.

[3] Xuejia Lai and James L. Massey, A Proposal for a New Block Encryption Standard, EUROCRYPT 1990, pp. 389–404

[4] Khovratovich D., Leurent G., Rechberger C. (2012) Narrow-Bicliques: Cryptanalysis of Full IDEA. In: Pointcheval D., Johansson T. (eds) Advances in Cryptology – EUROCRYPT 2012. EUROCRYPT 2012. Lecture Notes in Computer Science, vol 7237. Springer, Berlin, Heidelberg

[5] 2001. Performance Optimization of Numerically Intensive Codes. Soc. for Industrial and Applied Math., Philadelphia, PA, USA.

[6] Kelsey J., Schneier B., Wagner D. (1996) Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES. In: Koblitz N. (eds) Advances in Cryptology — CRYPTO '96. CRYPTO 1996. Lecture Notes in Computer Science, vol 1109. Springer, Berlin, Heidelberg

[7] Handschuh H., Heys H.M. (1999) A Timing Attack on RC5. In: Tavares S., Meijer H. (eds) Selected Areas in Cryptography. SAC 1998. Lecture Notes in Computer Science, vol 1556. Springer, Berlin, Heidelberg

[8] Massey J. SAFER K-64: A byte-oriented block-ciphering algorithm // Fast Software Encryption: Cambridge Security Workshop Cambridge, U. K., December 9–11,1993 Proceedings / R. J. Anderson — Berlin: Springer Berlin Heidelberg, 1994. — P. 1–17. — (Lecture Notes in Computer Science; Vol. 809) — ISBN 978-3-540-58108-6 — ISSN 0302-9743 — doi:10.1007/3-540-58108-1_1

[9] R. Rivest, A Description of the RC2(r) Encryption Algorithm (RFC2268, Informational), March 1998

[10] Simon and Speck: Block Ciphers for the Internet of Things (англ.). NIST Lightweight Cryptography Workshop (9 July 2015).

[11] RFC2994 A Description of the MISTY1 Encryption Algorithm. H. Ohta, M. Matsui. November 2000. (Format: TXT, HTML) (Status: INFORMATIONAL) (DOI: 10.17487/RFC2994)

[12] Ross Anderson, Eli Biham, Lars Knudsen. Serpent: A Proposal for the Advanced Encryption Standard

[13] Tom St Denis. Extended TEA Algorithms

[14] Ramanujam S., Karuppiah M. Designing an algorithm with high Avalanche Effect //IJCSNS International Journal of Computer Science and Network Security. – 2011. – Т. 11. – №. 1. – С. 106-111.