оставляет в этом направлении существенные перспективы. Атака на A5/1 методом «грубой силы» бесперспективна из-за большей, в сравнении с DES, длины ключа. Кроме этого, данная атака по эффективности более чем в 200 раз проигрывает известной SAT-атаке [3], реализация которой на GPU невозможна вследствие целого комплекса ограничений современных GPU-архитектур: отсутствие поддержки рекурсии, неэффективная обработка условных переходов, ограниченные возможности адресации памяти. Можно надеяться, что огромный интерес в мире к GPU-вычислениям позволит в ближайшее время в значительной степени преодолеть эти ограничения.

ЛИТЕРАТУРА

1. *Diffie W., Hellman M.* Exhaustive cryptanalysis of NBS data encryption standard // Computer. 1977. V. 10. P. 74–84.
2. *Hellman M.* A cryptanalytic time-memory trade-off // IEEE Trans. Inf. Theory. 1980. V. IT-26. P. 401–406.
3. *Kumar S., Paar C., Pelzl J., et al.* How to Break DES for Euro 8,980 // II Workshop on Special-purpose Hardware for Attacking Cryptographic Systems (SHARCS). Germany. 2006.
4. *Семенов А. А., Заикин О. С., Беспалов Д. В., и др.* Решение задач обращения дискретных функций на многопроцессорных вычислительных системах // Труды IV Междунар. конф. PACO'2008 (Москва, 26–29 октября 2008). 2008. С. 152–176.

УДК 519.7

ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА СКОРОСТИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ НЕКОТОРЫХ ПРЕОБРАЗОВАНИЙ, ИСПОЛЬЗУЮЩИХСЯ ПРИ СИНТЕЗЕ БЛОЧНЫХ КРИПТОСХЕМ

А. А. Дмух

Основными параметрами современных блочных криптосхем являются линейные преобразования, действующие на блоке, и нелинейные преобразования, представляющие собой параллельное применение нелинейных преобразований (подстановок) на блоках меньшей длины. В [1] указывается, что для обеспечения стойкости блочной криптосхемы линейные преобразования должны обладать достаточно хорошими рассеивающими свойствами. При этом основная трудоемкость одной итерации блочной криптосхемы, как правило, приходится на вычисление линейного преобразования.

При синтезе линейных преобразований разработчики пытаются достичь «золотой середины» между такими относительно противоречивыми требованиями к линейному преобразованию блочной криптосхемы, как максимизация скорости вычисления, минимизация памяти, необходимой для хранения, и достаточно хорошие рассеивающие свойства.

На данный момент в криптографической практике для реализации на ЭВМ одной итерации блочной криптосхемы используют подход (см., например, [2]), который заключается в использовании заранее вычисленных таблиц, в которых хранится результат композиции нелинейного и линейного преобразований. Результат применения одной итерации (после наложения ключа) представляет собой сумму по модулю 2 содержимого таблицы, взятого по байтам-адресам. Наилучшие показатели по скорости шифрования и количеству тактов процессора, необходимых для зашифрования одного байта, достигаются в случае, если таблица может быть целиком помещена в кэш-память процессора.

Однако с ростом n — длины блока — увеличивается и размер памяти, который необходим для хранения таблиц. Это приводит к двум негативным с точки зрения реализации последствиям. Во-первых, если криптосхема реализуется на вычислительных средствах с ограничениями по памяти (например, смарт-карты), то на длину блока накладываются существенные ограничения. Во-вторых, таблицы слишком больших размеров могут целиком не помещаться в кэш-память процессора, что существенно снижает скорость шифрования.

Пусть длина блока $n = m \cdot n' = k \cdot m' \cdot n'$, где n' — длина подстановки. В работе [3] предложено использовать линейные преобразования специального вида. В качестве линейного преобразования L рассматривается композиция $L(x) = A(P(x))$, где P представляет собой перестановку подвекторов длины n', $A = \mathrm{diag}(A_1, A_2, \ldots, A_k)$, $A_i = \overline{A}$, $i = 1, \ldots, k$, $\overline{A} \in GL(m' \cdot n', 2)$. При хранении в виде заранее вычисленных таблиц такое линейное преобразование занимает в k^2 раз меньше памяти, чем произвольное линейное преобразование.

Обычно на практике, за редким исключением, $n' = 8$, поэтому открытым остается вопрос, при каких значениях m' заранее вычисленные таблицы линейного преобразования целиком помещаются в кэш-память. Формально, для хранения таблиц необходимо $2^{n'}(m')^2 \cdot n'$ бит. Так как на ЭВМ помимо программы шифрования может быть запущено еще достаточно большое количество других приложений, то выполнение неравенства $2^{n'}(m')^2 \cdot n' < N_{\mathrm{cache}}$ (где N_{cache} — размер кэш-памяти процессора) является необходимым, но не достаточным условием полного размещения таблицы в кэш-памяти.

С целью определения наилучших с точки зрения скорости шифрования и количества тактов процессора, необходимых для обработки одного байта информации, значений $m' \in \{2, 4, 8, 16\}$ при различных значениях $k \in \{2, 3, \ldots, 8\}$ и $n' = 8$ и сопоставимом по криптографическим характеристикам числе итераций были проведены эксперименты, которые показали, что оптимальными значениями являются $m' = k = 2$, $m' = k = 4$ и $m' = k = 8$, которые соответствуют длине блока, равной 32, 128 и 512 бит соответственно. При данных значениях m' и k скорость шифрования в 2,5–1,3 раза выше, чем при любой другой фиксации.

ЛИТЕРАТУРА

1. *Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В.* Основы криптографии. М.: Гелиос АРВ, 2001. 201 с.
2. *Daemen J., Rijmen V.* AES Proposal: Rijndael. http://csrc.nist.gov — Document version 2, 1999.
3. *Rijmen V.* Cryptanalysis and Design of Iterated Block Ciphers. PhD, 2000.

УДК 519.7

О ТРУДОЁМКОСТИ НАПРАВЛЕННОГО ПЕРЕБОРА НЕРАВНОВЕРОЯТНЫХ ВАРИАНТОВ

И. В. Панкратов, О. А. Теплоухова

Решение криптографических задач часто состоит в определении значения ключевого параметра. Мощность множества K значений этого параметра может быть огромной. Кроме того, часто значение самого параметра может быть получено только методом опробования. Этот метод подразумевает последовательный перебор элементов в K до тех пор, пока не будет получено истинное значение. Бывает, что значения в K неравновероятны, и тогда можно осуществить их направленный перебор, начав