Д. В. Матюхин, В. А. Шишкин (Москва, ТВП)**. О криптографической стойкости функции хэширования ГОСТ Р 34.11-94.**

В работе [2] для хэш-функции ГОСТ Р 34.11-94 предложены алгоритмы построения прообраза с трудоемкостью 2^{192} вычислений функции сжатия (память 2^{70} байт) и коллизии с трудоемкостью 2^{105} вычислений функции сжатия, что меньше «универсальных» оценок 2^{256} и 2^{128} соответственно. В работе, представленной данным докладом, для алгоритма построения коллизии получена оценка объема памяти, предложена модификация алгоритма в условиях ограниченной памяти и получена оценка минимального объем памяти, при котором трудоемкость модифицированного алгоритма меньше «универсальной» оценки.

Обозначим $H_{i-1} = h_3\|h_2\|h_1\|h_0$ ($h_j \in \{0,1\}^{64}$) — промежуточное значение хэш-функции после обработки $i-1$-го блока сообщения, $M_i \in \{0,1\}^{256}$ — i-й блок сообщения, $S = s_3\|s_2\|s_1\|s_0$, $s_j = E(k_j, h_j)$ — результат зашифрования h_j на ключе k_j по ГОСТ 28147-89 в режиме простой замены, $k_j = F_j(H_{i-1}) \oplus G_j(M_i) \oplus c_j$, F_j, G_j — линейные преобразования, $c_j \in \{0,1\}^{256}$. В соответствии с ГОСТ Р 34.11-94, $H_i = \psi^{61}(H_{i-1} \oplus \psi(M_i \oplus \psi^{12}(S)))$, где ψ — линейное преобразование, или, эквивалентно, $\underbrace{\psi^{-74}(H_i)}_{X} = \underbrace{\psi^{-13}(H_{i-1})}_{Y} \oplus \underbrace{\psi^{-12}(M_i)}_{Z} \oplus S$, $X = x_3\|x_2\|x_1\|x_0$, $Y = y_3\|y_2\|y_1\|y_0$, $Z = z_3\|z_2\|z_1\|z_0$, где $x_j, y_j, z_j \in \{0,1\}^{64}$.

Объем памяти рассматриваемого алгоритма определяется способом построения коллизии функции $F(M_i) = x_3\|x_2\|x_1\colon \{0,1\}^{256} \to \{0,1\}^{192}$ (H_{i-1} — фиксированно) на блоках M_i с одинаковым значением x_0. В алгоритме каждый такой M_i равен $G_0^{-1}(k_0 \oplus F_0(H_{i-1}) \oplus c_0)$, где

$$k_0 = sk_0\|sk_1\|\ldots\|sk_7, \quad A_1(sk_0, sk_1, sk_2, sk_3)^T = d_1, \quad A_2(sk_4, sk_5, sk_6, sk_7)^T = d_2,$$

$sk_j \in \{0,1\}^{32}$, A_1, A_2 — матрицы над GF (2) размера 64×128, $d_1 + d_2$ фиксированно, $E(k_0, h_0) = h_0$. Для каждого значения d_1 находим 2^{64} значений k_0. Чтобы найти коллизию F, нужно в среднем порядка 2^{96} различных M_i, т. е. 2^{32} различных d_1. Тогда требуемый объем памяти составляет 2^{96} ячеек, в которые по адресу, состоящему из 12 байт двоичной записи $F(M_i)$, записываются остальные 12 байт и соответствующее d_1 (8 байт), итого, больше 2^{100} байт. Для сравнения: объем памяти наиболее мощного на сегодняшний день суперкомпьютера даже с учетом жестких дисков не превосходит 2^{54} байт [1].

Можно ли уменьшить объем памяти? Сделать это без увеличения трудоемкости алгоритма позволяют итерационные методы, но для их применения требуется легко вычисляемая функция $f\colon \{0,1\}^{192} \to \{0,1\}^{192}$, коллизия которой с большой вероятностью дает нужную коллизию F. Как построить такую f — не ясно. Однако, если память ограничена, можно модифицировать алгоритм ценой увеличения трудоемкости. А именно, пусть имеется память на 2^w ячеек по $192 - w + 64$ бита. Для каждого M_i в ячейку по адресу, состоящему из w младших бит $F(M_i)$, записываются остальные $192 - w$ бита вместе с соответствующим d_1. Если при повторном обращении к ячейке $192 - w$ битов совпали, то коллизия найдена, иначе перезаписываем содержимое ячейки.

В качестве вероятностной модели для оценки трудоемкости алгоритма естественно использовать выборку с возвращением из n элементов с памятью на z элементов [3]. В этой модели среднее число элементов, выбранных до первого повтора, при больших n можно считать равным $\sum_{k=0}^{z-1} e^{-k^2/(2n)} + (n/z)e^{-z^2/(2n)}$. Трудоемкость алгоритма определяется трудоемкостью построения коллизий для 512 различных F. В нашем случае $n = 2^{192}$, $z = 2^w$, поэтому средняя трудоемкость модифицированного

алгоритма приближенно равна

$$512\left(\sum_{k=0}^{2^w-1} e^{-k^2/2^{193}} + 2^{192-w}e^{-2^{2w-193}}\right)$$

вычислений функции сжатия. Ее легко вычислять, аппроксимируя сумму интегралом.

Можно показать, что средняя трудоемкость «универсального» метода построения коллизии хэш-функции ГОСТ Р 34.11-94 (параллельного метода поиска коллизий [3]) приближенно равна $3\sqrt{\pi}\,2^{128}$ вычислений функции сжатия. Вычисления показывают, что трудоемкость модифицированного алгоритма меньше этой величины при $w \geqslant 71$, т. е. памяти $2^w\lceil(192-w+64)/8\rceil > 2^{75}$ байт.

СПИСОК ЛИТЕРАТУРЫ

1. *Bland A. S.*, *Kendall R. A.*, *Kothe D. B.*, *Rogers J. H.*, *Shipman G. M.* Jaguar: The World's Most Powerful Computer. — In: CUG 2009 Proceedings. http://www.nccs.gov/wp-content/uploads/2010/01/Bland-Jaguar-Paper.pdf
2. *Mendel F.*, *Pramstaller N.*, *Rechberger C.*, *Kontak M.*, *Szmidt J.* Cryptanalysis of the GOST Hash Function. — Advances in Cryptology — CRYPTO 2008, p. 162–178.
3. *Oorschot van P. C.*, *Wiener M. J.* Parallel collision search with cryptanalytic applications. — J. Cryptology, 1999, v. 12, № 1, p. 1–28.