Об алгоритме выработки констант функции хэширования «Стрибог»

В.И. Рудской

Аннотация

В работе [2], посвященной криптографическому анализу нового отечественного стандарта функций хэширования ГОСТ Р 34.11-2012 («Стрибог»)[1], показано, что в конструкцию функции можно внедрить уязвимость путем специального выбора раундовых констант. К модифицированной таким образом функции хэширования возможно применение метода столкновений за практическое время. В подтверждение этих результатов, в работе [2] приводится набор раундовых констант и пара сообщений, которые образуют коллизию для модифицированной функции хэширования.

Для того чтобы снять подозрения о возможных заложенных уязвимостях в новом стандарте функций хэширования, в настоящей заметке мы приведем алгоритм, согласно которому были выбраны константы функций.

В настоящей заметке мы будем использовать обозначения стандарта ГОСТ Р 34.11-2012.

Существуют различные подходы к выбору констант, используемых в симметричных криптографических примитивах. Одним из возможных подходов является выбор констант случайным или псевдослучайным образом.

Криптографическая функция хэширования может рассматриваться как псевдо-случайная функция в том смысле, что хэш-значение не несет никакой информации о прообразе.

При разработке функции «Стрибог» было принято решение сгенерировать константы посредством вычисления значений «Стрибог»-подобной хэш-функции (которую мы обозначим через H_{init}) от различных входных сообщений. Эти сообщения и описание функции H_{init} представлены ниже.

Функция H_{init} практически идентична функции «Стрибог»-512 (которую мы обозначим через H), но имеет два небольших отличия. Первым отличием функции H_{init} от H является набор констант функции сжатия. Для H_{init} они были выбраны тривиальным образом, т.е. равными нулевым строкам: $C_1 = C_2 = \ldots = C_{12} = 0^{512}$.

Второе отличие H_{init} от H заключается в линейном преобразовании l пространства V_{64}, задаваемом в виде умножения справа на матрицу A над полем $GF(2)$. В функции H_{init} используется другая матрица A_{init}, причем A_{init} отличается от A обратным порядком следования *столбцов*: если $A = \{a_{i,j}\}_{i,j=0}^{63}$, то $A_{init} = \{a_{i,63-j}\}_{i,j=0}^{63}$. Легко убедиться, что, поскольку A является MDS-матрицей, то и заданная таким образом матрица A_{init} является MDS-матрицей.

В таблице 1 приведены сообщения M, которые были использованы для генерации раундовых констант функции хэширования.

$C_i = H_{init}(M)$	M (в шестнадцатеричной записи)
C_1	`e2e5ede1e5f0c3`
C_2	`f7e8e2eef0e8ece8e4e0ebc220e9e5e3f0e5d1`
C_3	`f5f3ecc4`
C_4	`f7e8e2eef0e4ede0f1eae5ebc020e9e5f0e4edc0`
C_5	`ede8e3fbc4`
C_6	`f7e8e2eeebe9e0f5e8cc20f1e8ede5c4`
C_7	`ede8f5fef2e0cc`
C_8	`f7e8e2eef0eef2eae8c220e9e8f0f2e8ecc4`
C_9	`e9eeeaf1e4f3d0`
C_{10}	`f7e8e2e5f0eee3c820f0e8ece8e4e0ebc2`
C_{11}	`ede8eaf8e8d8`
C_{12}	`f7e8e2e5e5f1eae5ebc020e9e8ebe8f1e0c2`

Таблица 1: Исходные сообщения, использованные для выработки констант функции хэширования

Список литературы

[1] ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования. — М. : ИПК Изд-во стандартов, 2012.

[2] *AlTawy, R.* Watch your Constants: Malicious Streebog / R. AlTawy, A. M. Youssef // *Cryptology ePrint Archive, Report 2014/879.* — 2014. — http://eprint.iacr.org/.