КРИТЕРИИ ПРЕДВАРИТЕЛЬНОЙ ОЦЕНКИ НАДЕЖНОСТИ НОВЫХ АЛГОРИТМОВ КРИПТОГРАФИЧЕСКОЙ ХЕШ-ФУНКЦИИ

Трунов Д.Н., магистр, d.n.trunov@gmail.com
КИИ ДонНТУ, Покровск, Украина

Криптографические хеш-функции в настоящее время широко применяются в системах электронной подписи, аутентификации пользователей, проверке парольных фраз [1], технологиях блокчейн (цепочки блоков транзакций) [2] и т.д. Причём с дальнейшим развитием и распространением таких технологий роль хеш-функций будет только возрастать, равно как и требования к их надёжности (криптографической стойкости).

В мире создано большое количество различных алгоритмов криптографических хеш-функций, наиболее известные из которых MD4, MD5, SHA-1 и ГОСТ Р 34.11-94. Однако работы по усовершенствованию существующих и созданию новых алгоритмов продолжаются, поскольку прогресс в развитии вычислительной техники позволяет находить слабые места в тех алгоритмах, которые когда-то считались вполне надёжными [3].

Разработкой таких алгоритмов занимаются не только профессиональные криптографы, но и энтузиасты-любители. Последние, ориентируясь на часто поверхностные описания требований к надёжным хеш-функциям, могут произвести алгоритм, который удовлетворяет этим требованиям только поверхностно. Как утверждает Брюс Шнайер: «Выглядящие совершенными криптосистемы часто оказываются чрезвычайно слабыми» [4].

Достаточно точно определить надёжность алгоритма можно только при помощи экспертной оценки и многочисленных проверок. Однако предварительную оценку можно дать и без участия экспертов по некоторым ключевым критериям, отличив заведомо ненадёжный алгоритм от потенциально надёжного.

При вычислении криптографических хеш-функций могут применяться комбинации следующих операций: ADD/SUB (сложение/вычитание), AND (логическое «И»), OR (логическое «ИЛИ»), XOR (исключающее ИЛИ), NOT (логическое отрицание), SHL/SHR (логический сдвиг влево/вправо), ROL/ROR (циклический сдвиг влево/вправо), MOD (остаток от деления) и другие. Чем меньше из этих операций участвуют в вычислении, тем менее надёжным может оказаться алгоритм.

Комбинация упомянутых операций не должна усекать пространство возможных результатов. Например, при вычислении функции A = (B SHL 3) AND (C SHL 5), число A никогда не будет принимать значение из диапазона 1..31, какими бы ни были значения B и C. Если подобное усечение в алгоритме присутствует, это даёт возможность создавать разные входные

блоки данных, для которых хеш-функция одинакова. В свою очередь, это нарушает одно из важнейших требований – стойкость к коллизиям [1, 4].

Важно, чтобы для каждого бита числа A вероятность принять значение 1 или 0 была равной 50%, поскольку некоторые комбинации операций смещают эту вероятность в ту или иную сторону. Например, если взять функцию A = (A AND B) AND C и выполнить её несколько раз в цикле, то вероятность того, что все биты числа A в конечном итоге будут равны 0, достигает почти 100% уже через несколько циклов, даже если на каждом шаге брать самые разные значения B и C. Эта ошибка ограничивает диапазон возможных результатов лишь косвенно, сильно уменьшая вероятность одних значений в пользу других.

Длина конечного результата хеш-функции в битах n должна быть достаточной, чтобы исключать как случайные коллизии, когда у двух разных блоков данных будет одинаковый хеш, так и противостоять умышленному поиску таких коллизий. Эффективность длины n можно оценить по величине общего времени перебора вариантов $T=t*2^n$, где t – время на вычисление одного варианта на конкретном компьютере. Если эта величина меньше, чем таковая для других распространённых алгоритмов, например, SHA-1, то алгоритм можно считать заведомо менее стойким. Но нужно помнить, что и распространённые алгоритмы устаревают.

Увеличение длины хеш-функции должно пойти на пользу её стойкости. Но если учесть ограничения на размер обрабатываемых чисел, то для увеличения длины функции нужно увеличивать количество чисел в работе. Часто такие числа обрабатываются небольшими группами отдельно друг от друга. Если эти группы никак или очень плохо связываются друг с другом при вычислении, то эффективная длина функции снизится до длины этой группы чисел. Отсюда и заниженная стойкость таких функций.

Ещё одним очень важным критерием является недопущение в алгоритме возможности вычисления функций по частям меньшей разрядности. Например, если есть функция A = Func1(A, B, C), где A, B, C – 32-разрядные числа и a, b, c – их части меньшей разрядности, то нужно исключить, чтобы для любой части существовала функция a = Func2(a, b, c). Иначе все значения функции Func2 можно вычислить заранее, поместив результаты в специальную таблицу. В дальнейшем несколько раз обратиться к таблице может быть проще, чем вычислять функцию Func1. Этот фактор может также существенно снизить стойкость алгоритма.

Дополнительным критерием может быть условие, чтобы при изменении любого бита входного сообщения, каждый бит результата изменился с вероятностью 50%. Для сообщений, состоящих из нескольких блоков, это условие должно работать для любого них. Если вероятность изменения каких-то битов результата существенно ниже 50%, это свидетельствует, скорее всего, о невыполнении одного или нескольких предыдущих критериев.

Таким образом, рассмотренные критерии позволяют при разработке нового алгоритма хеш-функции предварительно оценить степень его надёжности (криптографической стойкости), а также выявить и устранить наиболее типичные ошибки, приводящие к появлению уязвимостей в этом алгоритме.

Литература

1. Википедия. Криптографическая хеш-функция. [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/Криптографическая_хеш-функция

2. Википедия. Цепочка блоков транзакций. [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/Цепочка_блоков_транзакций

3. Xiaoyun Wang, Yiqun Lisa Yin, Hongbo Yu. Finding Collisions in the Full SHA-1 [Электронный ресурс] Режим доступа: http://www.infosec.sdu.edu.cn/uploadfile/papers/Finding%20Collisions%20in%20the%20Full%20SHA-1.pdf

4. Брюс Шнайер. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Триумф, 2002. – 816 с.

Анотація

Розглянуто критерії, що доповнюють і уточнюють вимоги до криптографічно стійких хеш-функцій. Показано можливість попередньої оцінки надійності нових алгоритмів.

Ключові слова: криптографічна стійкість, хеш-функція, алгоритм.

Аннотация

Рассмотрены критерии, дополняющие и уточняющие требования к криптографически стойким хеш-функциям. Показана возможность предварительной оценки надёжности новых алгоритмов.

Ключевые слова: криптографическая стойкость, хеш-функция, алгоритм.

Abstract

The article describes the criteria for supplementing and clarifying requirements for cryptographically resistant hash functions. The possibility of a preliminary assessment of the reliability of new algorithms is shown.

Keywords: resistance cryptographic, hash function, algorithm.