УДК 004.056.5

ХЕШИРОВАНИЕ ДАННЫХ, ПРИНЦИП РАБОТЫ

Жизнев Д.И., Волков Б.С., Тодорико Е.С.
Новокаховский политехнический колледж ОНПУ, УКРАИНА

АННОТАЦИЯ. В статье рассмотрено принцип функционирования процесса хеширования данных и актуальные виды криптографических хеш-функций.

Введение. На сегодняшний день криптографические хеш-функции – этонезаменимый инструмент, который используется для проведения операций аутентификации, обеспечения целостности данных и их защиты. Хеширование используется повсеместно, будь то сайт или прикладная программа. Владеть навыками шифрования данных должен каждый специалист по информационной безопасности [1].

Цель работы. Описать процесс хеширования данных. Охарактеризовать наиболее применяемые на сегодняшний день хеш-функции.

Основная часть работы. Хеширование данных – это процесс преобразования строки произвольной длины в строку фиксированной длины, используя алгоритм хеш-функции. Другими словами, хеширование «сжимает» входные данные и дает на выходе уникальную строку определенной длинны. Данные на выходе обычно называют хешем или хеш-суммой.

Существует ряд определенных свойств, которыми должна обладать хеш-функция, чтобы считаться безопасной:

1. Детерминированние. В независимости от количества раз, которые была прохеширована входная строка через одну и ту же хеш-функцию, результат будет один и тот же. Это важно, потому что если вы будете получать разные хеши каждый раз, будет невозможно отслеживать ввод.
2. Быстрая скорость вычислений. Из-за медленной скорости обработки входной строки, система будет элементарно неэффективна.
3. Сложность обратного вычисления.
4. Коллизионная устойчивость. Это значит, что хеш-функция не должна преобразовывать разные сообщения в одинаковый хеш. Чем выше этот параметр, тем меньше шансов у злоумышленников расшифровать исходный текст. Так же под устойчивостью к коллизиям понимают сложность нахождения пары сообщений с одинаковыми значениями свертки. Обычно именно нахождение способа построения коллизий криптоаналитиками служит первым сигналом устаревания алгоритма и необходимости его скорой замены [2].

С точки зрения математики, хеш-функцией называется всякая функция:

$$h: X \to Y, \tag{1}$$

где X – множество всех сообщений, Y–множество двоичных векторов фиксированной длины,а для любого сообщения Mзначение Hимеет фиксированную длину:

$$h(M) = H \tag{2}$$

Как правило, хеш-функции строят на основе так называемых одношаговых сжимающих функцийдвух переменных:

$$y = f(x_1, x_2), \tag{3}$$

где x_1, x_2 и y – двоичные векторы длины m, n и m соответственно, причем m–длина блока сообщения, а n – длина свертки.

Для получения значения H входная строка разбивается на блоки длиной m (если длина сообщение не кратна m, то последний блок разбавляется так называемой «криптографической солью» – строкой случайных данных, таким образом блок дополняется до полной длины). Затем к блокам $M_1, M_2, \ldots, M_n$ применяют процедуру вычисления свертки:

$$H_0 = v$$
$$H_i = f(M_i, H_{i-1}),\ i = 1,\ldots,N \qquad (4)$$
$$h(M) = H_N.$$

где v – инициализирующий вектор (может представлять собою любые единичные или набор случайных данных, например, выборка из даты и времени).

Также стоит отметить, что свойства хеш-функции будут определены свойствами одношаговой сжимающей функцией, благодаря которой она была создана. Рассмотрим ситуацию, когда два пользователя доверяют друг другу, но высока вероятность перехвата данных посторонним пользователем. В таком случае, нужно использовать ключевые хеш-функции (коды аутентификации). Они гарантируют, без применения дополнительных средств, правильность источника данных (сообщение отправлено одним из доверенных пользователей) и целостность данных (данные не были перехвачены и изменены).

В другой ситуации, когда пользователи не доверяют друг другу и есть вероятность перехвата, то используются бесключевые хеш-функции, которые гарантируют при использовании дополнительных средств (шифрование, к примеру) целостность данных [3]. С конца XX века начали пользоваться популярностью такие криптографические алгоритмы:

MD4 – разработан в 1990 году Рональдом Ривестом, профессором Массачусетского университета. Функция генерирует 128-разрядное сообщение. MD4 использовалась в протоколе MS-CHAP, который разработан корпорацией Microsoft для определение подлинности удаленных рабочих станций на базе ОС Windows. Состоит из 5 шагов, которые формируют одну операцию. Хеширование при помощи этого алгоритма состоит из 48 операций.

MD5 – разработан в 1991 году тем же профессором из Массачусетского университета. Использует 128-битное сообщение. Широко применялся для проверки целостности информации и хранение хешей паролей. На протяжение первого десятилетия XXI века были найдены критические ошибки в функции, которые привели к запрету использование алгоритма в 2008 году в связи с ненадежностью. В 2011 году опубликован официальный документ, который признает MD5 ненадежным и призывает отказаться от его использования.

SHA-1 – алгоритм хеширования, который генерирует 160-битный хеш. Создан в 1995 году параллельно с MD4 Рональдом Ривестом. Алгоритм использует хеш-функцию, основанную на функции сжатия, описанной выше. Используется и в наше время, в США рекомендован как основной к использованию в государственных учреждениях.

SHA-2 – семейство криптографических алгоритмов, разработано в 2005 году АНБ США. Включает в себя алгоритмы SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/256 и SHA-512/224, SHA-1. Используется в приложениях, связанных с защитой информации. Работает на основе структуры Меркла-Дамгарда (алгоритм разбивает входное сообщение на блоки и производит с ними операции).

SHA-3 или Keccak – алгоритм хеширования переменной длины, разработан в 2008 году группой ученых, во главе которых был Йоанн Даймен, который до этого занимался разработкой личных алгоритмов хеширования. Работает по принципы криптографической губки, который обеспечивает большую защищенность данных, по сравнению с SHA-1 и SHA-2 [4].

Выводы. Хеш-функции и процесс хеширования в целом представляют собой интересный механизм для выполнения различных операций по защите данных. Их грамотное использование предотвращает перехват ценной информации и обеспечивает надежную передачу данных. Современные криптографические хеш-алгоритмы совершенствуют эти процессы, а также сводят к минимуму появления коллизий – основного источника обхода таких алгоритмов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. KasperskyLabDaily. Чудеса хеширования. [Електронний ресурс]. – Режим доступу: URL: https://www.kaspersky.ru/blog/the-wonders-of-hashing/3633/

2. Хабрахбр. Что такое хэширование? [Електронний ресурс]. – Режим доступу: URL: https://habrahabr.ru/post/345740/

3. Хабрахабр. Хэш-алгоритмы. [Електронний ресурс]. – Режим доступу: URL: https://habrahabr.ru/post/93226/