

03, сентябрь 2018

УДК 004.056.2

Применение алгоритмов хеширования, для обеспечения целостности информации, хранящейся в облачных структурах

Назарова Е.К., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационная безопасность»
alenan95@mail.ru*

*Научный руководитель: Медведев Н.В., доцент, к.т.н.
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационная безопасность»
iu8@bmstu.ru*

Аннотация: В статье анализируются алгоритм хеширования данных в облачных структурах, обеспечивающий требуемый уровень защищенности данных. Приведены примеры работы алгоритма MD5. Рассмотрен приём, позволяющий защитить пароли от раскрытия с помощью баз данных распространённых хешей.

Ключевые слова: облачные технологии, защита информации, целостность информации, хеширование, MD5.

Введение

При хранении информации в облачных структурах, нужно понимать, что данные всегда будут доступны через интернет. Разумеется, такая ситуация благоприятствует деятельности хакеров. Так что к безопасности и целостности информации в облачных структурах нужно подходить особенно щепетильно. Помочь в этом сумеет современный алгоритм шифрования данных – хеширование.

Криптографической хеш-функцией, называют функцию вида: $y=f(x)$ которая удовлетворяет следующим свойствам:

1. На вход хеш-функции может поступать последовательность данных произвольной длины, результат же (называемый хеш, или дайджест) имеет фиксированную длину.

2. Значение y по имеющемуся значению x вычисляется за полиномиальное время, а значение x по имеющемуся значению y почти во всех случаях вычислить невозможно.
3. Вычислительно невозможно найти два входных значения хеш-функции, дающие идентичные хеши.
4. При вычислении хеша используется вся информация входной последовательности.
5. Описание функции является открытым и общедоступным.

Для авторизации пароль, введённый пользователем, нужно сравнивать с паролями, которые находятся на сервере в базе данных. Однако хранение их в открытом виде может привести к тому, что злоумышленник при помощи SQL-инъекции просто похитит все пароли. Поэтому для защиты крайне ценной информации применяется хеширование.

Хеширование – особый способ шифрования данных, при котором исходная строка произвольной длины превращается в строку определённой длины. То есть если ввести единицу или абзац текста, то на выходе получится 2 строки одинаковой длины, но с разными символами. Функция, которая выполняет шифровку и расшифровку, называется хеш-функцией.

Если применить простое шифрование с прямой заменой символов, то злоумышленник, добавляя по одному символу к концу строки, сумеет разгадать алгоритм изменения пароля.

Хеш-функция может при шифровании двух разных строк возвращать одинаковый хеш. Такая неприятная ситуация называется коллизией. Шанс появления коллизии является главной характеристикой хеш-функции. Чем ниже вероятность, тем лучше.

Имеются различные способы хеширования данных, которые основаны на различных математических операциях. В идеале процедура не должна давать коллизий. Однако на практике такие алгоритмы не встречаются. Поэтому для снижения вероятности коллизий используется не одна хеш-функция, а несколько, конкретная выбирается по случайному алгоритму.

Рассмотрим один из наиболее распространённых алгоритмов хеширования данных в облачных структурах – MD5 (с длиной хеша 128 бит). Он изменяет любое количество символов в строку из 32 шестнадцатеричных цифр. На рисунке 1 представлена схема работы алгоритма MD5.

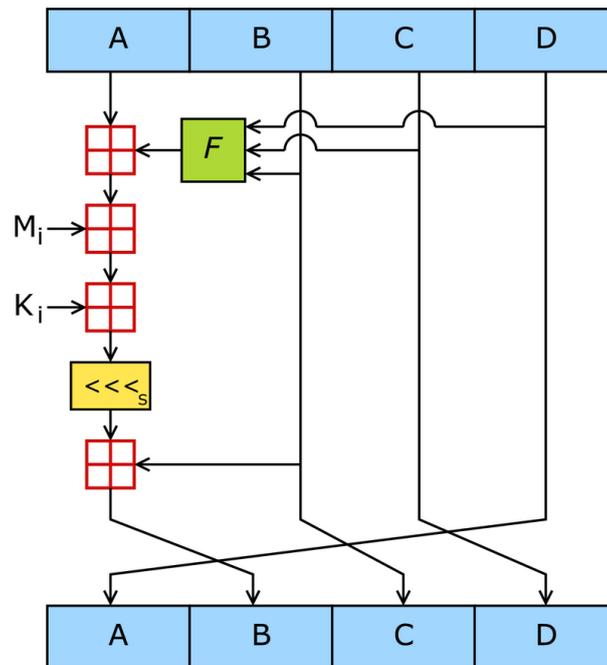


Рис. 1. Схема работы алгоритма MD5

Разберём несколько примеров работы алгоритма хеширования MD5:

1. Если захешировать строку «md5» – MD5("md5") – то функция вернёт следующее: 1BC29B36F623BA82AAF6724FD3B16718.
2. При этом если внести даже минимальное изменение, то числовой ряд полностью поменяется. Например, напишем «md4», тогда функция MD5("md4") вернёт такой ряд чисел: C93D3BF7A7C4AFE94B64E30C2CE39F4F.
3. Даже если ввести пустую строку, то есть написать функцию MD5(""), то в итоге всё равно получится число из 32 шестнадцатеричных цифр: D41D8CD98F00B204E9800998ECF8427E.

Такая особенность, когда замена одного символа полностью меняет хеш, именуется лавинным эффектом. Это крайне важное свойство, так как без него злоумышленник может по получаемому хешу догадаться об алгоритме изменения данных.

Проблема в том, что функция MD5 довольно популярна, так что хеши многих распространённых паролей можно найти в интернете. То есть если хакер обнаружит пароли, то сумеет добыть пароль, который соответствует этому хешу. Частично решить такую проблему можно с помощью двойного хеширования, вот так: MD5(MD5("пароль")). Однако даже хеши от данного приёма стали достаточно распространены.

Поэтому в настоящий момент используется приём «соления». «Соль» – строка из нескольких символов, которая добавляется к концу пароля перед пропуском сквозь хеш-функцию. Такой приём позволит защитить пароли от раскрытия с помощью баз данных распространённых хешей.

Также, в совокупности, применяют метод контрольных сумм. Под контрольной суммой следует понимать набор символов, который ни за что не отвечает, а является маркером целостности данных. То есть при передаче сообщения посредством хеширования может произойти неконтролируемое изменение информации. Такая ошибка будет сразу видна, когда передаваемые данные необъёмные, к тому же утилиту можно сразу проверить на практике.

Если хешируется большая программа, тогда ошибка будет видна далеко не сразу, что является серьёзным минусом. Проверить целостность переданной информации сумеет контрольная строка из различных символов. Если она будет одинакова до хеширования и после, то, скорее всего, целостность данных не пострадала.

Заключение

В данной статье был проанализирован алгоритм хеширования, который имеет длину хеша 128 бит, а именно алгоритм MD5. Были рассмотрены примеры работы данного алгоритма. Для защиты от атак на облачное хранилище с целью нарушения целостности информации, хранящейся в облачных структурах, необходимо применение алгоритмов хеширования.

Список литературы

- [1]. Клементьев И.П., Устинов В.А. Введение в облачные вычисления // Интернет университет информационных технологий. Режим доступа: <http://www.intuit.ru/departement/se/incloudc/> (дата обращения: 22.10.2016).
- [2]. Коммуникации для бизнеса. Надежна ли защита у вашего Internet-провайдера. Режим доступа: <http://www.osp.ru> (дата обращения: 12.05.2018).
- [3]. Машкина И.В., Рахимов Е.А., Васильев В.И. Методика построения модели комплексной оценки угроз информации, циркулирующей на объекте информатизации // Известия ТРТУ. Материалы VIII научно-практической конференции «Информационная безопасность». Таганрог: ТРТУ. 2006. С. 70-76.
- [4]. Мельников В.В. Безопасность информации в автоматизированных системах. М.: Финансы и статистика. 2003. 368 с.
- [5]. Gentry C. A fully homomorphic encryption scheme / C. Gentry - Stanford University, Ph.D. thesis. 2009.
- [6]. Li J., Wang Q., Wang C., Cao N., Ren K., Lou W. Fuzzy keyword search over encrypted data in cloud computing. Mini-Conf. IEEE INFOCOM. 2010. 5 p. DOI: 10.1109/INFOCOM.2010.5462196

- [7]. Miller R. Who Has the Most Web Servers? 2012. Режим доступа: <http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers/> (дата обращения: 14.05.2018).
- [8]. R.L. Rivest, L. Adleman, M.L. Dertouzos. On data banks and privacy homomorphisms // Foundations of secure computation. 1978. Vol. 32. No. 4. Pp. 169–178.