Содержание:

Введение

Криптографические хеш-функции — незаменимый и повсеместно распространенный инструмент, используемый для выполнения целого ряда задач, включая аутентификацию, проверку целостности данных, защиту файлов и даже обнаружение зловредного ПО. Существует масса алгоритмов хеширования, отличающихся криптостойкостью, сложностью, разрядностью и другими свойствами. Считается, что идея хеширования принадлежит сотруднику IBM, появилась около 50 лет назад и с тех пор не претерпела принципиальных изменений. Зато в наши дни хеширование обрело массу новых свойств и используется в очень многих областях информационных технологий.

1. Что такое хеш-функция и как она работает?

Если коротко, то криптографическая хеш-функция, чаще называемая просто хешем, — это математический алгоритм, преобразовывающий произвольный массив данных в состоящую из букв и цифр строку фиксированной длины. Причем при условии использования того же типа хеша длина эта будет оставаться неизменной, вне зависимости от объема вводных данных. Криптостойкой хеш-функция может быть только в том случае, если выполняются главные требования: стойкость к восстановлению хешируемых данных и стойкость к коллизиям, то есть образованию из двух разных массивов данных двух одинаковых значений хеша. Интересно, что под данные требования формально не подпадает ни один из существующих алгоритмов, поскольку нахождение обратного хешу значения — вопрос лишь вычислительных мощностей. По факту же в случае с некоторыми особо продвинутыми алгоритмами этот процесс может занимать чудовищно много времени.

Например, мое имя — Brian — после преобразования хеш-функцией SHA-1 (одной из самых распространенных наряду с MD5 и SHA-2) при помощи онлайн-

генератора будет выглядеть так: 75c450c3f963befb912ee79f0b63e563652780f0. Как вам скажет, наверное, любой другой Брайан, данное имя нередко пишут с ошибкой, что в итоге превращает его в слово brain (мозг). Это настолько частая опечатка, что однажды я даже получил настоящие водительские права, на которых вместо моего имени красовалось Brain Donohue. Впрочем, это уже другая история. Так вот, если снова воспользоваться алгоритмом SHA-1, то слово Brain трансформируется в строку 97fb724268c2de1e6432d3816239463a6aaf8450. Как видите, результаты значительно отличаются друг от друга, даже несмотря на то, что разница между моим именем и названием органа центральной нервной системы заключается лишь в последовательности написания двух гласных. Более того, если я преобразую тем же алгоритмом собственное имя, но написанное уже со строчной буквы, то результат все равно не будет иметь ничего общего с двумя предыдущими: 760e7dab2836853c63805033e514668301fa9c47.

Впрочем, кое-что общее у них все же есть: каждая строка имеет длину ровно 40 символов. Казалось бы, ничего удивительного, ведь все введенные мною слова также имели одинаковую длину — 5 букв. Однако если вы захешируете весь предыдущий абзац целиком, то все равно получите последовательность, состоящую ровно из 40 символов: c5e7346089419bb4ab47aaa61ef3755d122826e2. То есть 1128 символов, включая пробелы, были ужаты до строки той же длины, что и пятибуквенное слово. То же самое произойдет даже с полным собранием сочинений Уильяма Шекспира: на выходе вы получите строку из 40 букв и цифр. При всем этом не может существовать двух разных массивов данных, которые преобразовывались бы в одинаковый хеш.

2. Сферы применения хеш-функции и как при помощи неё ловить вирусы.

Отличный вопрос. Однако ответ не так прост, поскольку криптохеши используются для огромного количества вещей.

Для нас с вами, простых пользователей, наиболее распространенная область применения хеширования — хранение паролей. К примеру, если вы забыли пароль к какому-либо онлайн-сервису, скорее всего, придется воспользоваться функцией восстановления пароля. В этом случае вы, впрочем, не получите свой старый пароль, поскольку онлайн-сервис на самом деле не хранит пользовательские

пароли в виде обычного текста. Вместо этого он хранит их в виде хеш-значений. То есть даже сам сервис не может знать, как в действительности выглядит ваш пароль. Исключение составляют только те случаи, когда пароль очень прост и его хеш-значение широко известно в кругах взломщиков. Таким образом, если вы, воспользовавшись функцией восстановления, вдруг получили старый пароль в открытом виде, то можете быть уверены: используемый вами сервис не хеширует пользовательские пароли, что очень плохо.

Вы даже можете провести простой эксперимент: попробуйте при помощи специального сайта произвести преобразование какого-нибудь простого пароля вроде «123456» или «password» из их хеш-значений (созданных алгоритмом MD5) обратно в текст. Вероятность того, что в базе хешей найдутся данные о введенных вами простых паролях, очень высока. В моем случае хеши слов «brain» (8b373710bcf876edd91f281e50ed58ab) и «Brian» (4d236810821e8e83a025f2a83ea31820) успешно распознались, а вот хеш предыдущего абзаца — нет. Отличный пример, как раз для тех, кто все еще использует простые пароли.

Еще один пример, покруче. Не так давно по тематическим сайтам прокатилась новость о том, что популярный облачный сервис Dropbox заблокировал одного из своих пользователей за распространение контента, защищенного авторскими правами. Герой истории тут же написал об этом в твиттере, запустив волну негодования среди пользователей сервиса, ринувшихся обвинять Dropbox в том, что он якобы позволяет себе просматривать содержимое клиентских аккаунтов, хотя не имеет права этого делать.

Впрочем, необходимости в этом все равно не было. Дело в том, что владелец защищенного копирайтом контента имел на руках хеш-коды определенных аудио- и видеофайлов, запрещенных к распространению, и занес их в список блокируемых хешей. Когда пользователь предпринял попытку незаконно распространить некий контент, автоматические сканеры Dropbox засекли файлы, чьи хеши оказались в пресловутом списке, и заблокировали возможность их распространения.

Где еще можно использовать хеш-функции помимо систем хранения паролей и защиты медиафайлов? На самом деле задач, где используется хеширование, гораздо больше, чем я знаю и тем более могу описать в одной статье. Однако есть одна особенная область применения: хеширование широко используется для детектирования зловредных программ защитным ПО.

Примерно так же, как звукозаписывающие лейблы и кинопрокатные компании защищают свой контент, сообщество создает списки зловредов (многие из них доступны публично), а точнее, списки их хешей. Причем это может быть хеш не всего зловреда целиком, а лишь какого-либо его специфического и хорошо узнаваемого компонента. С одной стороны, это позволяет пользователю, обнаружившему подозрительный файл, тут же внести его хеш-код в одну из подобных открытых баз данных и проверить, не является ли файл вредоносным. С другой — то же самое может сделать и антивирусная программа, чей «движок» использует данный метод детектирования наряду с другими, более сложными.

Криптографические хеш-функции также могут использоваться для защиты от фальсификации передаваемой информации. Иными словами, вы можете удостовериться в том, что файл по пути куда-либо не претерпел никаких изменений, сравнив его хеши, снятые непосредственно до отправки и сразу после получения. Если данные были изменены даже всего на 1 байт, хеш-коды будут отличаться, как мы уже убедились в самом начале статьи. Недостаток такого подхода лишь в том, что криптографическое хеширование требует больше вычислительных мощностей или времени на вычисление, чем алгоритмы с отсутствием криптостойкости. Зато они в разы надежнее.

Кстати, в повседневной жизни мы, сами того не подозревая, иногда пользуемся простейшими хешами. Например, представьте, что вы совершаете переезд и упаковали все вещи по коробкам и ящикам. Погрузив их в грузовик, вы фиксируете количество багажных мест (то есть, по сути, количество коробок) и запоминаете это значение. По окончании выгрузки на новом месте, вместо того чтобы проверять наличие каждой коробки по списку, достаточно будет просто пересчитать их и сравнить получившееся значение с тем, что вы запомнили раньше. Если значения совпали, значит, ни одна коробка не потерялась.

Заключение

Хеширование, которое родилось еще в середине прошлого века, активно используется в наши дни везде, где требуется произвести быструю выборку данных. Появились новые методы хеширования, новые модификации алгоритмов, написанных ранее. По мнению Дональда Кнута, наиболее важным открытием в области хеширования со времен 70 годов, вероятно, является линейное хеширование Витольда Литвина. Линейное хеширование, которое не имеет ничего

общего с классической технологией линейной адресации, позволяет многим хеш-адресам расти и/или выступать в поли вставляемых и удаляемых элементов. Линейное хеширование может также использоваться для огромных баз данных, распределенных между разными узлами в сети.

Разумеется, методы и сферы применения хеширования не ограничиваются тем, что представлено в этой работе. Не вдаваясь в строгий анализ эффективности, были рассмотрены только базовые, наиболее известные методы.

Целью работы является:

— Понять что такое хеш-функция и рассмотреть принцип её работы;

В ходе выполнении данной выпускной квалификационной работы были поставлены и следовательно выполнены следующие задачи:

Произведен анализ сфер применения хеш-функции;

Произведено рассмотрение методов хеширования данных.

Список литературы

1. https://infourok.ru/kriptograficheskaya-hesh-funkciya-3774965.html.
2. https://znanio.ru/media/informatika-2578792.
3. https://4mak.ru/tehnologii/hesh-chto-eto-takoe-i-kak-hesh-funktsiya-pomogaet-reshat-voprosy-bezopasnosti-v-internete.html.
4. https://moluch.ru/young/archive/28/1666/.
5. https://meridianinvest.ru/na-zametku/kriptograficheskie-hesh-funkcii.html.