Доклад на тему ‘Хэш-функции’

Дисциплина: Основы информационной безопасности

Явкина Анастасия Юрьевна, НПМбд-02-21

Содержание

1 Введение

Хеширование — это преобразование информации с помощью особых математических формул. В результате возникает хеш (hash) — отображение данных в виде короткой строки, в идеале — уникальной для каждого набора информации. Размер строки может быть одинаковым для информации разного объема.

Хеш — это не зашифрованная исходная информация. Это скорее уникальная метка, которая генерируется для каждого набора данных индивидуально. Хеш состоит из цифр и латинских букв. Если захешировать большую книгу и одно слово, получатся хеши одинаковой длины. А если изменить в слове одну букву и снова захешировать полученную строку, новый хеш будет совершенно другим, там не окажется участков, которые повторяли бы предыдущий.

2 Глава 1. Хэш-функция.

Хеш-функция — это математический алгоритм, по которому хешируется информация. Его название тоже иногда сокращают как «хеш». Хеш-функций существует очень много, они различаются методами вычислений, назначением, надежностью и другими параметрами. Можно сказать, что хеш-функцией называют алгоритм, который преобразует входные данные произвольной длины в выходные данные фиксированной длины. Длина хеша бывает разной — 64, 128 или 256 бит — значение зависит от типа хеш-функции.

2.1 Глава 1.1 Применение хэш-функций.

Основное назначение хеширования — проверка информации. Эта задача важна в огромном количестве случаев: от проверки паролей на сайте до сложных вычислений в блокчейне. Так как хеш — это уникальный код определенного набора данных, по нему можно понять, соответствует ли информация ожидаемой. Поэтому программа может хранить хеши вместо образца данных для сравнения. Это может быть нужно для защиты чувствительных сведений или экономии места.

Вот несколько примеров:
• вместо паролей на сервере хранятся хеши паролей;
• антивирус хранит в базе хеши вирусов, а не образцы самих программ;
• электронная подпись использует хеш для верификации;
• информация о транзакциях криптовалюты хранится в виде кешей;
• коммиты в Git идентифицируются по хешу

Среди других, менее распространенных примеров использования — поиск дубликатов в больших массивах информации, генерация ID и построение особых структур данных. Это, например, хеш-таблицы — в них идентификатором элемента является его хеш, и он же определяет расположение элемента в таблице.

Глава 1.2 Свойства криптографических хеш-функций.

1. Необратимость. Из хеша нельзя получить исходные данные даже теоретически. Слишком много информации отбрасывается в процессе; это не зашифровка информации.
2. Детерминированность. Если подать хеш-функции одинаковые данные, то и хеш у них будет одинаковым. Именно это свойство позволяет использовать хеши для проверки подлинности информации.
3. Уникальность. Идеальная хеш-функция выдает стопроцентно уникальный результат для каждого возможного набора данных. В реальности такое невозможно, и иногда случаются коллизии — одинаковые хеши для разных сведений. Но существующие хеш-функции достаточно сложны, поэтому вероятность коллизии сводится к минимуму.
4. Разнообразие. Даже если два набора информации различаются одним-двумя символами, их хеши будут кардинально разными. У них не будет общих блоков, по ним невозможно будет понять, что исходные данные схожи.
5. Высокая скорость генерации. Это в целом свойство любых хешей: в отличие от зашифрованных версий файлов, они генерируются быстро, даже если входной массив данных

3 Глава 2. Принцип работы хэш-функции.

Возможных преобразований для получения хеша бесконечное количество. Это могут быть формулы на основе умножения, деления и других операций, алгоритмы разного уровня сложности.

К примеру:
"Хеш-функции", основанные на делении:
«Хеш-код» как остаток от деления на число всех возможных «хешей»
Хеш-функция может вычислять «хеш» как остаток от деления входных данных на M:
$h(k)=k \bmod M$
где M— количество возможных «хешей» (выходных данных). При чётном M и при чётном k значение функции будет чётным. При чётном M и при нечётном k значение функции будет нечётным. Не следует использовать в качестве M степень основания системы счисления

компьютера, так как «хеш-код» (выходные данные) будет зависеть только от нескольких цифр числа k (входных данных), расположенных справа, что приведёт к большому количеству коллизий. На практике в качестве M обычно выбирают простое число; в большинстве случаев этот выбор вполне удовлетворителен.

Но если хеш применяется для защиты данных, его функция должна быть криптографической — такие хеш-функции обладают определенными свойствами. Именно криптографические хеш-функции используются, например, при хранении паролей.

Если говорить о криптографической хеш-функции, то она чаще всего работает в несколько шагов. Данные разбиваются на части и проходят через сжимающую функцию, которая преобразовывает информацию в меньшее количество бит. Функция должна быть криптостойкой — такой, результат которой практически невозможно вскрыть.

А вот хеш-функции для более простых случаев, например построения таблиц, не обязаны быть криптографическими. Там преобразования могут быть проще.

Глава 3. Безопасность криптографической хеш-функции.

Цель использования хешей — обеспечить безопасность пользователей. Идентификация или проверка подлинности данных нужны, чтобы никто не мог воспользоваться чувствительной информацией в своих целях. Поэтому специалисты пользуются именно криптографическими хеш-функциями. Они должны быть безопасными — так, чтобы никто не мог взломать их.

Идеальная криптографическая хеш-функция полностью отвечает перечисленным ниже требованиям. Реальные не могут ответить им на 100%, поэтому задача их создателей — максимально приблизиться к нужным свойствам.

1. Стойкость к коллизиям. коллизия — явление, когда у двух разных наборов данных получается одинаковый хеш. Это небезопасно, потому что так злоумышленник сможет подменить верную информацию неверной. Поэтому коллизий стремятся максимально избегать.

Современные криптографические хеш-функции не полностью устойчивы к коллизиям. Но так как они очень сложные, для поиска коллизии нужно огромное количество вычислений и много времени — годы или даже столетия. Задача такого поиска становится практически невыполнимой.

2. Стойкость к восстановлению данных. Частично это означает все ту же необратимость. Но

восстановить данные в теории можно не только с помощью обратной функции — еще есть метод подбора. Стойкость к восстановлению данных подразумевает, что, даже если злоумышленник будет очень долго подбирать возможные комбинации, он никогда не сможет получить исходный массив информации. Это требование выполняется для современных функций. Информации в мире настолько много, что полный перебор всех возможных комбинаций занял бы бесконечно большое количество времени.

3. Устойчивость к поиску первого и второго прообраза. Первый прообраз — как раз возможность найти обратную функцию. Такой возможности нет, ведь криптографическая хеш-функция необратима. Этот пункт пересекается с требованием стойкости к восстановлению данных.\ Второй прообраз — почти то же самое, что нахождение коллизии. Разница только в том, что в случае со вторым прообразом ищущий знает и хеш, и исходные данные, а при поиске коллизии — только хеш. Хеш-функция, неустойчивая к поиску второго прообраза, уязвима: если злоумышленник будет знать исходные данные, он сможет подменить информации.

4 Заключение

Таким образом, хэш-функции — это важный инструмент в арсенале компьютерных технологий. Они играют ключевую роль в обеспечении безопасности данных, проверке целостности информации и оптимизации вычислений. Важно выбирать хэш-функции с учетом их надежности и области применения. Современные криптографические хэш-функции, такие как SHA-256 и SHA-3, продолжают оставаться надежными и востребованными.

5 Список литературы

1. Винокуров, С.Ф. Избранные вопросы теории булевых функций / С.Ф. Винокуров. - М.: [не указано], 2012. - 564 с.
2. Дэвенпорт, Дж. Интегрирование алгебраических функций / Дж. Дэвенпорт. - М.: [не указано], 2013. - 726 с.
3. Евграфов, М.А. Асимптотические оценки и целые функции / М.А. Евграфов. - М.: [не указано], 2014. - 285 с.
4. Катленд, Н. Вычислимость. Введение в теорию рекурсивных функций / Н. Катленд. - М.: [не указано], 2011. - 707 с.
5. Ленг, С. Введение в алгебраические и абелевы функции / С. Ленг. - М.: [не указано], 2014. - 601 с.
6. Мальгранж, Б. Идеалы дифференцируемых функций / Б. Мальгранж. - М.: [не указано], 2011. - 874 с.
7. Мальцев, А.И. Алгоритмы и рекурсивные функции / А.И. Мальцев. - М.: [не указано], 2016. - 629 с.
8. Марченков, С.С. Замкнутые классы булевых функций: моногр. / С.С. Марченков. - М.: [не указано], 2015. - 946 с.
9. Рассел, Джесси Tiger (хэш-функция) / Джесси Рассел. - М.: VSD, 2012. - 833 с.
10. Шевалле, К. Введение в теорию алгебраических функций от одной переменной / К. Шевалле. - М.: [не указано], 2014. - 226