Кулаков Алексей, 244 группа

Реферат

Коллизия хеш-функции

Для того, чтобы хеш-функция *H* считалась криптографически стойкой, она должна удовлетворять трем основным требованиям:

- Необратимость: для заданного значения хеш-функции *m* должно быть практически невозможно найти блок данных *X*, для которого *H*(*X*) = *m*.
- Стойкость к коллизиям первого рода: для заданного сообщения *M* должно быть практически невозможно подобрать другое сообщение *N*, для которого *H*(*N*) = *H*(*M*).
- Стойкость к коллизиям второго рода: должно быть практически невозможно подобрать пару сообщений (M, M'), имеющих одинаковый хеш.

Использование коллизий для взлома

В качестве примера можно рассмотреть простую процедуру аутентификации пользователя:

- при регистрации в системе пользователь вводит свой пароль, к которому применяется некоторая хеш-функция, значение которой записывается в базу данных;
- при каждом вводе пароля, к нему применяется та же хеш-функция, а результат сравнивается с тем, который записан в БД.

При таком подходе, даже если злоумышленник получит доступ к базе данных, он не сможет восстановить исходные пароли пользователей (при условии необратимости используемой хеш-функции). Однако, если злоумышленник умеет находить коллизии для используемой хеш-функции, ему не составит труда найти поддельный пароль, который будет иметь ту же хеш-сумму, что и пароль пользователя.

Можно использовать коллизии для подделки сообщений: информация о валютных операциях, к примеру, часто шифруется посредством хеш-функций; злоумышленник, обладая методом нахождения коллизий этой хеш-функции, может заменить сообщение поддельным и тем самым повлиять на ход валютной операции.

Схожим образом можно использовать коллизии для подделки цифровых подписей и сертификатов.

Защита от использования коллизий

Существует ряд методов защиты от взлома, защиты от подделки паролей, подписей и сертификатов, даже если злоумышленнику известны методы построения коллизий для какой-либо хеш-функции.

Одним из методов является метод «salt», применяемый при хранении UNIX-паролей — добавление некоторой последовательности символов перед хешированием. Иногда, эта же последовательность добавляется и к полученному хешу. После такой процедуры, итоговые хеш-таблицы значительно сложнее анализировать, а так как эта последовательность секретна, существенно повышается сложность построения коллизий — злоумышленнику должна быть также известна последовательность «salt».

Другим методом является конкатенация хешей, получаемых от двух различных хеш-функций. При этом, чтобы подобрать коллизии к хеш-функции $C(x) = y(x)\|z(x)$, являющейся конкатенацией хеш-функций $y(x)$ и $z(x)$, необходимо знать методы построения коллизий и для $y(x)$, и $z(x)$. Недостатком конкатенации является увеличение размера хеша, что не всегда приемлемо в практических приложениях.

Методы поиска коллизий

Одним из самых простых и универсальных методов поиска коллизий является атака «дней рождения». С помощью этой атаки отыскание коллизии для хеш-функции разрядности n бит потребует в среднем около $2^{n/2}$ операций. Поэтому n-битная хеш-функция считается криптостойкой, если вычислительная сложность нахождения коллизий для нее близка к $2^{n/2}$.

Кроме того, существует атака расширения: зная $H(x)$, можно вычислить $H(x\|y) = H(H(x)\|y)$; которая, для некоторых хеш-функций, работает даже при обеспечении стойкости к коллизиям первого рода, стойкости к коллизиям второго рода, а также свойства необратимости. Подразумевается, что нет необходимости знать X, а достаточно знать лишь его хеш. Таким образом можно, например, дописывать дополнительную информацию к чужому сообщению. Для предотвращения этой атаки используют различные методы: добавляют дополнительный раунд при хешировании, отличный от предыдущих; применяют многократное хеширование; или используют комбинацию предыдущих 2х методов.

Но атаку расширения можно рассмотреть и с другой стороны: если у нас есть некоторое сообщение X, и хэш-функция уязвима к атаке расширения, то легко можно найти коллизию первого рода — $M_1 = X\|Y$, $M_2 = H(X)\|Y$, $H(M_1) = H(M_2)$, то есть нарушается свойство стойкости к коллизиям первого рода.

Большая часть современных хеш-функций имеют одинаковую структуру, основанную на разбиении входного текста на блоки и последующем итерационном процессе, в котором на каждой итерации используется некоторая функция $G(x,y)$, где x — очередной блок входного текста, а y — результат предыдущей операции. Однако такая схема несовершенна, так как, зная функцию G, мы можем проводить анализ данных в промежутках между итерациями, что облегчает поиск коллизий.