



Курс: эксплуатация подсистем безопасности АС

Тема: **Хэш-функция**

Преподаватель: Пятков
Антон Геннадьевич

Красноярск

Определение и свойства

Хеш-функция $H(x)$ — функция, которая преобразует (отображает) сообщение произвольной длины в число («свёртку») фиксированной длины.

Хеш-функция должна обладать следующими свойствами.

1. Хеш-функция может быть применена к аргументу любого размера.
2. Выходное значение хеш-функции имеет фиксированный размер.
3. Хеш-функцию $H(M)$ достаточно просто вычислить для любого M (простота вычисления образа).
4. Для любого y с вычислительной точки зрения невозможно найти x , такое что $H(x) = y$ (сложность вычисления прообраза).
5. Для любого фиксированного x с вычислительной точки зрения невозможно найти z , не равное x , такое, что $H(x) = H(z)$ (стойкость к коллизиям, вычислению второго прообраза).

Для криптографической хеш-функции (в отличие от хеш-функции общего назначения) сложно вычислить обратную и даже найти два сообщения с общей хеш-свёрткой.

По 4 свойство $H(x)$ - односторонняя функция, поэтому $H(x)$ можно использовать в качестве контрольной суммы для проверки целостности.

Суть хэш-функции

Функции хэширования используют алгоритмические примитивы (циклы, условия) и по внутренним преобразованиям делятся на:

- функции, использующие битовые логические преобразования (побитовые нелинейные операции "И", "ИЛИ", "НЕ", "НЕ ИЛИ", сдвиги);
- функции, использующие блочные симметричные шифры;
- функции, использующие преобразования в группах, полях и кольцах с целочисленным или полиномиальным базисом;
- функции, использующие матричные преобразования.



Области использования

Области использования хеш-функции:

- ✓ защита паролей при их передаче и хранении;
- ✓ формирование контрольных кодов MDC (Manipulation Detection Code) - кода обнаружения манипуляций с данными;
- ✓ получение сжатого образа сообщения перед формированием электронной подписи;
- ✓ задачи поиска данных.

ГОСТ Р 34.11-2012 «Стрибог»

ГОСТ Р34.11-2012 - криптографический стандарт вычисления хеш-функции.

Размер блока входных данных: 512 бит

Размер хеша (хэш-свёртки): 512 бит



Стандарт определяет алгоритм и процедуру вычисления хеш-функции для последовательности символов.

Стандарт обязателен для применения в качестве алгоритма хеширования в государственных организациях РФ и ряде коммерческих организаций.

ГОСТ Р 34.11-2012 «Стрибог»

Основная операция функции сжатия обозначается как LPS (3 вида преобразований: подстановки на байтах, транспонирования матрицы байт и умножения 64-битных векторов на матрицу 64×64 в $GF(2)$):

1. S — нелинейная биекция. 512 бит аргумента рассматриваются как массив из шестидесяти четырёх байт, каждый из которых заменяется по заданной стандартом таблице подстановки;
2. P — переупорядочивание байт. Байты аргумента меняются местами, транспонирование матрицы байт размером 8×8 ;
3. L — линейное преобразование. Аргумент рассматривается как 8 64-битных векторов, каждый из которых заменяется результатом умножения на определённую стандартом матрицу 64×64 над $GF(2)$.

В функции сжатия используются только преобразование LPS и побитовое исключающее ИЛИ над 512-битными блоками. Вместе со сложением по модулю 2^{512} они составляют полный набор операций, использующихся в функции хеширования ГОСТ Р 34.11-2012.

Значение функции сжатия на каждом шаге зависит от предыдущего шага
=> невозможно обрабатывать блоки одного потока данных параллельно.

Другие хэш-функции

- ✓ LM-хеш (стандартная хэш-функция в ОС Windows);
- ✓ NT-хеш (стандартная хэш-функция в ОС Windows);
- ✓ MD2;
- ✓ MD4;
- ✓ MD5;
- ✓ MD6;
- ✓ SHA-1;
- ✓ SHA-2;
- ✓ Кецсак (SHA-3);
- ✓ ГОСТ Р 34.11-94;
- ✓ Прочие: CubeHash, BLAKE, BMW, ECHO, Fugue, JH, Hamsi, HAVAL, Luffa, N-Hash, RIPEMD-128, RIPEMD-160, RIPEMD-256, RIPEMD-320, Scrypt, SHABAL, SHAvite-3, SIMD, Skein, Snefru, Tiger, Whirlpool и др.

Принцип проверки парольной фразы

Хранить пароли открыто нецелесообразно, т.к. при НСД к хранилищу паролей они станут известны злоумышленнику.

Можно хранить не пароли, а лишь их хэш-свёртки.

Пусть дана парольная фраза `pass`.

Создадим свёртку $H(\text{pass})$ и сохраним на сервере

Вариант простого общения Клиент-Сервер:



Клиент, К

Сервер, С

1. Вычисляет значение $H(\text{pass})$
2. Отправка на сервер $H(\text{pass})$
2. Сверяет полученное и хранимое значение

Перехватив все сообщения между К и С, криптоаналитик может восстановить пароль, взломав хеш-функцию.

Атака на хэш-функцию

Для вскрытия паролей, преобразованных при помощи сложнообратимой хеш-функции, а также для атак на симметричные шифры на основе известного открытого текста используются радужные таблицы.

Радужная таблица – специальный вариант таблиц поиска для обращения криптографических хеш-функций.

Радужная таблица – готовая построенная цепочка возможных паролей.

! таблицы могут взламывать только ту функцию, для которой создавались

! использование функции выведения ключа с применением соли делает эту атаку неосуществимой.

Принцип проверки парольной фразы

Пусть дана парольная фраза $pass$.

Создадим свёртку, но с дополнительной солью – числом $R2$ (заранее выбранное псевдослучайное число). Вычисляем и храним $H(pass, R2)$

Вариант улучшенного общения Клиент-Сервер:

Клиент, K

Сервер, C

1. Выбор псевдослучайного $R1$
2. Отправка на сервер $(name, R1)$
3. Посылает значение $R2$
4. Вычисляет значение $H(R1, H(pass, R2))$
 5. Вычисляет значение $H(R1, H(pass, R2))$
6. Отправка на сервер $H(R1, H(pass, R2))$
7. Сверяет полученные и вычисленное

Перехватив все сообщения между K и C , криптоаналитик теоретически не может восстановить пароль, передаваемое хеш-значение каждый раз разное и время на ответ ограничено.