Алгоритм хэширования Keccak.

Батарин Егор

Аннотация

Статья посвящена обзору самому передовому на данный день алгоритму хеширования - Keccak. Рассмотрены история его создания, определение хэш-функции, основные свойства криптографических хэш-функций, общая структура алгоритма Keccak.

1 История создания алгоритма

В 2012 году сообщество NIST провело конкурс алгоритмов SHA-3, после того, как были проведены успешные атаки на предыдущие алгоритмы семейства SHA. На конкурс были приняты 51 алгоритмов, в результате естественного отбора которых в качестве нового стандарта SHA был принят алгоритм Keccak.

2 Что такое хэш-функция?

Хэш-функция - это отображение, которое ставит в соответствие строке произвольной длины строку фиксированной длины. Как правило, мощность отображаемого множества больше мощности множества значений хэш-функции, а значит она не может быть инъективной. Пара сообщений, нарушающих инъективность функции, называется коллизией.

Сферы применения хэш-функций различны:

- Построение уникального идентификатора для данного набора данных
- Для вычисления контрольных сумм и обнаружения ошибок
- При сохранении пароля в виде хэш-кода
- При выработке электронной подписи на хэш-код

Последние два применения взяты из криптографии. Не любые хэш-функции годятся для применения в этих случаях. Для этого нужно использовать специальные хэш-функции, называемые криптографическими, которые обеспечивают повышенную безопасность системы.

3 Свойства криптографических хэщ-функций

Чтобы хэш-функция была годна для защиты информации, она должна обладать следующими свойствами:

-Сопротивление поиску первого прообраза: для данного хэш-кода трудно найти исходное сообщение, которое было отображено данной хэш-функцией в данный хэш-код. Иными словами, для данной хэш-функции трудно построить функцию, обратную к ней.

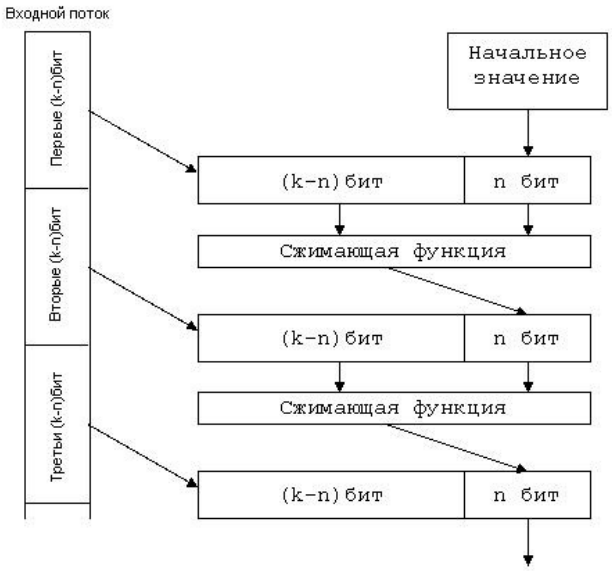
-Сопротивление поиску второго прообраза: для данного сообщения трудно найти другое сообщение, отображаемое хэш-функцией в тот же хэш-код.

-Стойкость к коллизиям: нет эффективного полиномиального алгоритма, который находит коллизии.

-Лавинный эффект: малое изменение входных значений хэш-функции сильно меняет выходное значение.

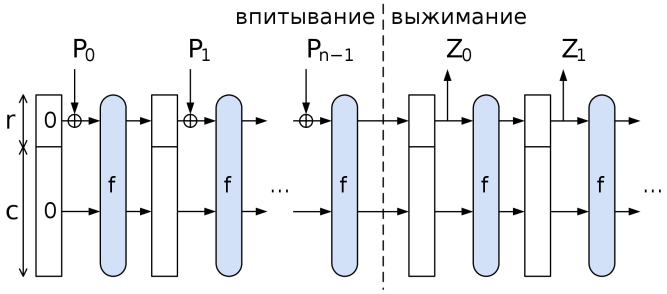
Как правило, криптографические хэш-функции строятся по итеративной последовательной схеме.

Такая схема позволяет достичь лавинного эффекта, поскольку каждый бит выходного потока зависит от всего входного потока данных. В основе распространенной схеме лежит разделение входного сообщения на множество блоков, каждый из которых обратывается сжимающей функцией. Затем полученное значение используется для последующих итераций. Для того, чтобы длина входного сообщения была кратна длине блока, его удлинняют дополнительными битами до нужной длины. Похожие идеи используются в алгоритме Keccak. Такая констукция называется губкой.



4 Губка

Вычисление хэш-кода в констукциях по типу губки происходит в несколько стадий.



1) Исходное сообщение M дополняется до строки P так, чтобы длина последнией была кратна r. Для этого используется функция дополнения.

2) Строка P делится на строки P_0,...,P_{n-1}.

3) Происходит этап "впитывания":

3.1) Каждый блок P_i дополняется нулями до строки длины b бит - верхние входы над вертикальными стрелочками на рисунке

3.2) Генерируется начальная строка состояния S длины $b = r + c$, заполненная нулями - самый левый белый прямоугольник на рисунке.

3.3) Она суммируется по модулю 2 вместе со строкой P_0, а результат обрататывается функцией перестановок f.

3.4) Полученная в результате действия f строка длины b суммируется по модулю 2 с P_1 и все последующие операции повторяются итеративно.

4) Происходит этап "отжимания": на этом этапе суммирование по модулю 2 прекращается и к строке состояния применяется функция перестановок. К конечному результату Z каждый раз добавляются первые r бит строки состояние. Происходит это ддо того момента, пока длина Z будет не меньше d. Когда это достигнуто, лишняя часть обрезается и в конце концов получается строка Z длины d.

5 Функция перестановок f

Как мы видели выше, строка состояния S претерпевает изменения под действием функции перестановок f в результате "впитывания"и "отжимания". Авторы представляют эту строку как массив $5 \times 5 \times 64$, стало быть $A[i][j][k]$ - это $(5i + j) \times 64 + k$ бит строки S.

Действие функции f разбивается на 3 шага:

1. Перевод S в A

2. Итеративное действие $A_i = \iota_i(\chi(\pi(\rho(\theta(A)))))$, где функция ι_i зависит от номера раунда i, итерация применяется по раундам

3. Перевод конечного массива A' на шаге 2 в строку S' длины b

6 Список литературы

1. https://dx.doi.org/10.6028%2Fnist.fips.202
2. https://doi.org/10.6028/NIST.SP.800-185
3. https://ru.wikipedia.org/wiki/SHA-3
4. https://habr.com/ru/post/534082/
5. https://ru.wikipedia.org/wiki/Хеш-функция
6. https://ru.wikipedia.org/wiki/Функция_губки