

1. Intro

Hash function F that transforms arbitrary length data into fixed size data(digest, hash).

Practically output range is 128-512 bits

Requirements:

0. Function is fast and have small memory consuming
1. one wayness. Computational infeasible to find x from $y=h(x)$. It's map.
- 1.1 value distribution is equal 2^{-n}
2. weak collision resistance(first kind). Computational infeasible to find x_2 from $y=h(x_1)=h(x_2)$
3. strong collision resistance(second kind). Computational infeasible to find and x_1, x_2 from $y=h(x_1)=h(x_2)$

Main target: infeasible to find collision.

The key s generally not keep secret, nevertheless, H_s must be resistant to collisions.

2. Formal definition

A hash function H_s with fixed-length output $l(n)$ is a pair of probabilistic polynomial time algorithms (Gen, H) satisfying the following:

Gen is a probabilistic algorithm that takes as input a security parameter $1^n(1111\dots 1_n)$ and outputs a key s .

2.1. Difficult or Computational infeasible

Not solvable in asymptotic polynomial time.

2.2. Preimage resistance

Hash function must be strength to find preimage of hash.

Use cases:

- find hashed password by brute force

2.3. weak collision(second preimage resistance)

Given $y = h(x_1)$, computationally infeasible to find $x_2 : y = h(x_2)$

Use cases:

- fake signature

2.4. strong collision

Computationally infeasible to find $x_2, x_1 : y = h(x_2)=h(x_1)$

Use cases:

- find two documents with the single hash

Requires to compute $2^{(N/2)}$ to find x_2 and x_1 .

3. Birthday problem

In set of n randomly chosen people, to get the probability of two has same birthday 50%+ required only 23 people.

$$\begin{aligned} \text{no overlap at all } P_0 &= 1 * \left(\frac{365-1}{365}\right) * \left(\frac{365-2}{365}\right) \dots * \left(\frac{365-i}{365}\right) \\ \text{at least 1 overlap } P_1 &= 1 - P_0 \end{aligned}$$

For 23 people

$$P_0 = 0.4972 \longrightarrow P_1 = 0.5028$$

Another proof: n people

$$P(1) = 1 - P_0,$$

$$P_0 = \frac{V_{\text{no pair}}}{V_{\text{all}}}$$

$$V_{\text{no pair}} = P_{365}^n = \frac{(365)!}{(365-n)!}$$

$$V_{\text{all}} = 365^n$$

$$P_0 = \frac{P_{365}^n}{365^n} = \frac{(365)!}{(365-n)!365^n}$$

$$n = 23 \rightarrow P_0 \sim 50\%$$

“whoop”

Permutation count of rearrangement combinations. The number of permutations n is

$$P_n = n!$$

Partial permutation count of rearrangement combination of subset k elements from set n .

$$P_n^k = \frac{n!}{(n-k)!}$$

Combination is a k -element subset of s , the elements in combination are not ordered. ($k!$ means number of permutations in each k -length subset of S)

$$C_n^k = \frac{n!}{(n-k)!k!}$$

4. Based on block ciphers

- MD4
- MD5
- SHA-1
- SHA-2

4.1. Block cipher

Block cipher function that operates on fixed bits length input. Input n , key k . Output n size message.

Standard block cipher: AES, DES.

4.1.1. AES

Symmetric cipher. Key size 128/192/256. $N = 128$

Rounds will depend of the key size. DES has fixed 16 rounds.

Each round is derived into layers First round turns into two sub keys and 4 layers. Rest of the rounds, one key per time, 3 layers.

3 types of layers.

1. Key addition layer.(XOR with key)
2. Byte substitution layer(S-box): perform substitution using “lookup tables”. Provide confusion
3. Diffusion layer:
 - ShiftRows: permutes the data on the byte level
 - MixColumn: another matrix permutation

4.1.2. XOR

Usage justification: it's better randomizes encryption, since it output is 0/1 50%

4.2. Blockchain

Sequential growing data structure, intended to provide complete data integrity.

Mining problem = for H function and fixed k , find x that $H(x)$ starts with k nulls.

4.3. Use cases

- Hash table(often used non-cryptographic hash functions) and indexing
- Fingerprinting and verifying the integrity of data
- Identifier

5. Merkle–Damgård construction

domain extension method

Def a method of building collision-resistant cryptographic hash functions from collision-resistant one-way compression functions.(uses AES(state, message))

If compression F is resistant to collisions \rightarrow construction is resistant too.

Sequential compression of blocks(like blockchain) if end is not full length, add padding.

It's possible to process as a tree - therefore scales infinitely (called merkle tree).

5.1. Other ciphers

Stream cipher encrypts data bit by bit. Useful for real time data processing.