1. Into

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: malicious adversary cannot replace or modify data without changing it digest. Function should have behavior like random function.

1.1. Difficult or Computational infeasible

Not solvable in asymptotic polynomial time.

1.2. Preimage resistance

Hash function must be strength to find preimage of hash.

Use cases:
- find hashed password by brute force

1.3. weak collision(second preimage resistance)

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

Use cases:
- fake signature

1.4. 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.

2. Birthday problem

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

$$\text{no overlap at all } P_0 = 1 * \left(\frac{365-1}{365}\right) * \left(\frac{365-2}{365}\right) ... * \left(\frac{365-i}{365}\right)$$

$$\text{at least 1 overlap } P_1 = 1 - P_0$$

For 23 people

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

Another proof: n people

$P(1) = 1 - P_0,$

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

$$V_{\text{no\_pair}} = P^n_{365} = \frac{(365)!}{(365-n)!}$$

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

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

$$n = 23 \to P_0\text{\textasciitilde}50\%$$

"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.

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

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)

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

3. Based on block ciphers

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

3.1. 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.

3.1.1. 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 dirived 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

3.1.2. XOR

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

4. Merkle–Damgård construction

4.1. Use cases

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