Intro to One-way Hash Functions

Overview & Properties of Hash Functions

Overview of One-way Hash Functions

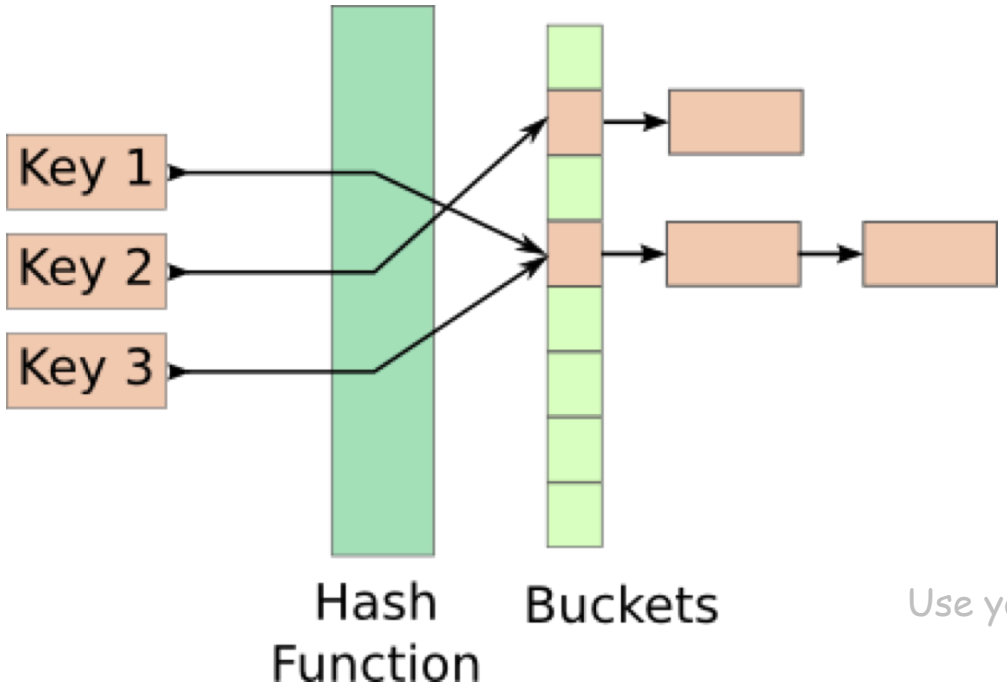
- One-way Hash Functions are an essential building block in cryptography, with desirable **practical** and **security** properties.

- Applications
  -> integrity verification, password authentication, commitments, etc.

- Possible Attacks
  -> collision attacks, length extension attacks

Hash Functions (and Hash Tables)

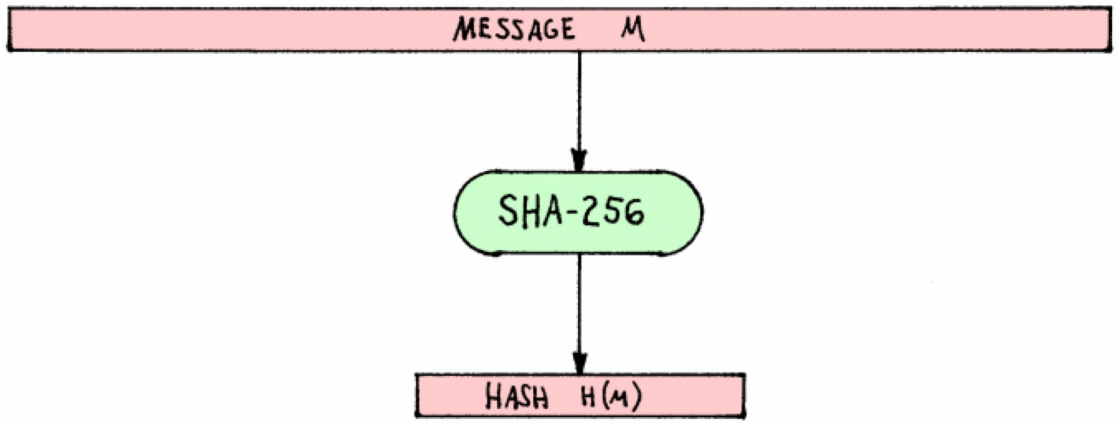
- Difference from "Normal" Hash Function
  - Hash function: maps arbitrary size data to data of fixed size
  - Example: f(x) = x mod 100



Collisions happen…
Use your favorite collision resolution technique (open addressing, chaining, etc.)

Practical Properties of One-Way Hash Functions

- **Compression**: compress **arbitrarily long inputs** into **fixed-length outputs**

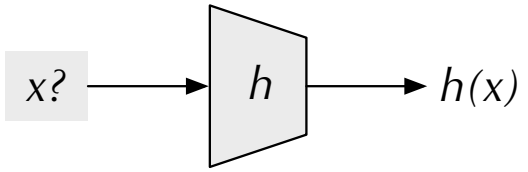


- **Easy to compute**: fast and easy (speed + efficiency) to compute

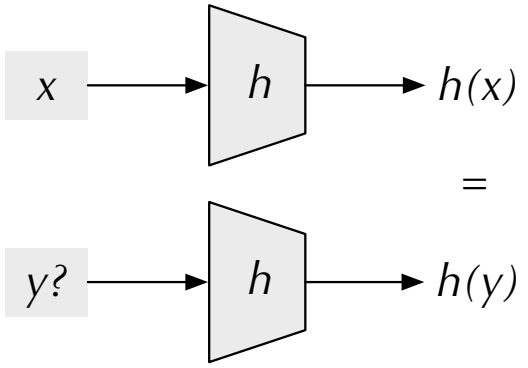
```
$ openssl speed
Doing md5 for 3s on 256 size blocks: 5123210 md5's in 3.00s
Doing hmac(md5) for 3s on 256 size blocks: 4907417 hmac(md5)'s in 3.00s
Doing sha1 for 3s on 256 size blocks: 5720106 sha1's in 2.99s
Doing sha256 for 3s on 256 size blocks: 3289471 sha256's in 3.00s
Doing sha512 for 3s on 256 size blocks: 2248701 sha512's in 3.00s
```

Security Properties of One-Way Hash Functions

- **Preimage Resistance ("One-Way")**
  Given h(x) = z, hard to find x
  (or any input that hashes to z for that matter)



- **Second Preimage Resistance**
  Given x and h(x), hard to find y s.t. h(x) = h(y)



- Collision Resistance (or, ideally, "Collision Free")
  Difficult to find x and y s.t. hash(x) = hash(y)