Hashing

Introduction to Basic Cryptography

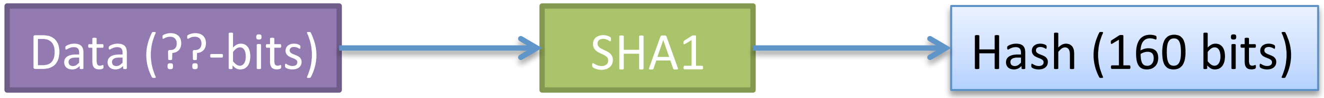
Dr. Ryan Riley

جامعة قطر
QATAR UNIVERSITY

Intro

- Take an arbitrary message, compute a fixed length hash
- Sometimes called a message digest
- Used outside of security as well
  - Not all hash functions can be used for security
  - We are concerned with *cryptographic hash functions*

Example



Sample in Python

```
import hashlib
m = hashlib.sha1()
m.update("Hi there, I want to hash this")
m.update("I want this to be hashed, too")
d = m.digest()
d.encode("hex")
```

Properties of a Secure Hash Function

1. Pre-image resistance (or One Way)
   – Infeasible to determine M from H(M)
2. Second pre-image resistance
   – Given M_1, infeasible to find M_2 such that $H(M_1) = H(M_2)$
3. Collision resistance
   – Can't find *any* M_1, M_2 such that $H(M_1) = H(M_2)$

Breaking Pre-Image Resistance

- Given a hash, find a message with the same hash
- Bruteforce approach: Pick a message, hash it, compare to the hash you have
- How long will this take?
  - Best case: First guess is correct! (1)
  - Worst case: You find all others first (2^{128} - 1)
  - Average case: You find it halfway through (~2^{128}/2 = 2^{127})

Breaking 2nd Pre-Image Resistance

- Attack is basically the same as breaking pre-image resistance

Breaking Collision Resistance

- Things get more complicated
- You need to learn a part of probability called the *birthday paradox*

Birthday Collision

Assuming all birthdays are equally likely, how many people do I need to get into a room before two of them have the same birthday? (Let’s call this a birthday collision)

Birthday Paradox

- Rule of thumb: If there are N different possibilities of something, then you need sqrt(N) randomly chosen items in order to have a 50% chance of a collision
    - In the birthday example, sqrt(365) ~= 23
    - You need ~23 random people to have a 50% chance of a birthday collision

Birthday Paradox and Hashing

- Recall collision resistance: “Can’t find *any* M_1, M_2 such that $H(M_1) = H(M_2)$”

- How many hashes do I need to collect before a hash collision occurs?
  - For a 128-bit hash, there are 2^{128} possible hashes, so applying the birthday paradox…
    - sqrt(2^{128}) ~= 2^{64}
  - Still a big number, but this means the strength is similar to breaking pre-image resistance for a 64-bit hash…

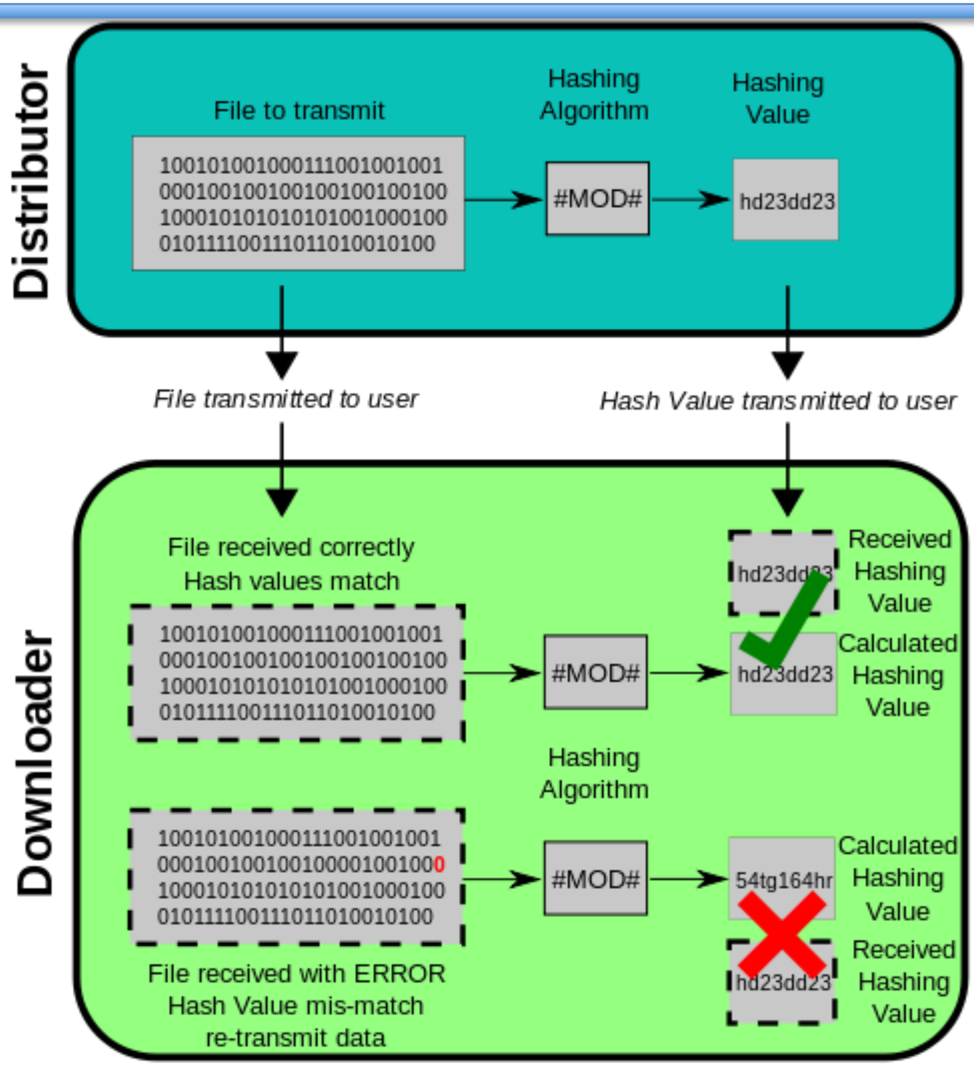
Examples of Real Hash Functions

- MD5
  - Produces a 128-bit hash
  - Collisions can be found in ~2^21 hashes
- SHA1
  - 160-bit hash
  - Collisions can be found in 2^61 hashes
- SHA2
  - Actually 4 different hash functions: SHA-224, SHA-256, SHA-384, SHA-512
  - Minor attacks, but still good
- SHA3
  - Just chosen as a new NIST standard
  - No known attacks

Applications of Hash Functions

- Detect errors in file transfers
  - BitTorrent does this
- Message Authentication Code (MAC)
- Password storage
- More!

Application: File Transmission



Application: MAC

- Hashing with a key. The goal is to create a hash that can only be created or verified by someone with the key
- Different techniques
  - H(m | K)
    - Bad because of how some hash functions are designed
  - H(K | m)
    - Better
  - H(K | m | K)
    - Better still
  - H(K | H(K | M))
    - Provably good. (But slower)

Application: Password Storage

- When designing an application that stores passwords, don't store them in plaintext
  - If someone steals your password file, then they have all the user passwords!
  - Store hashes instead
  - Note: If you really are going to do this, don't just store hashes. Read about something called PBKDF

```
103238726-|--|-tanman_127@hotmail.com-|-mv7OR0Hbks/ioxG6CatHBw==-

103238727-|--|-dadangahmad-|-7WkoOEfwfTTioxG6CatHBw==-

103238728-|--|-lingbo5426@yahoo.cn-|-clpn0KbcrWbioxG6CatHBw==-

103238729-|--|-raganaxi_tony@hotmail.com-|-bSU1JVB9CaI5IQsp4TdDow==-

103238730-|--|-jilliec2005@gmail.com-|-F0uvI/LK8wpbbW05Qn4LHQ==-

103238731-|--|-sfernand@ucsc.edu-|-w5lqfGenk2vioxG6CatHBw==-
```

Summing Up

- Hash functions take an arbitrary message, compute a fixed length hash
- Have many applications in computer science