BLM5106- Advanced Algorithm Analysis and Design

H. İrem Türkmen

<u>Introduction to algorithms</u> **TH Cormen, CE Leiserson, RL Rivest, C Stein**

Hash Functions

A hash function h maps arbitrary strings of data to fixed length output. The function is deterministic and public, but the mapping should look "random". In other words,

$$h : \{0,1\}^* \to \{0,1\}^d$$

for a fixed d. Hash functions do not have a secret key. Since there are no secrets and the function itself is public, anyone can evaluate the function. To list some examples,

Hash function	MD4	MD5	SHA-1	SHA-256	SHA-512
d	128	128	160	256	512

Hash Functions

- In practice, hash functions are used for “digesting” large data. For example, if you want to check the validity of a large file (potentially much larger than a few megabytes), you can check the hash value of that file with the expected hash.
- That is, it should be “hard” to find two inputs m1 and m2 for hash function h such that h(m1)= h(m2).

Desirable Properties

- One-way (pre-image resistance)
- Collision-resistance (Strong collision-resistance)
- Target collision resistance (Weak collision-resistance)
- Pseudo-random
- Non-malleability

One-way (pre-image resistance)

- Given $y \in \{0,1\}^d$,it is hard to find an x such that h(x)= y.
- Assume h(x) =x mod p, is h(x) one way?

Collision-resistance

- It is hard to find any pair of inputs x, x' such that x<>x’ and $h(x) = h(x')$
- Is it important for storing passwords?
- What about a look up tabe?

Target collision resistance

- Given x, it is hard to find x' such that x<>x’ and h(x)= h(x')

Pseudo-random

- The function behaves indistinguishable from a random oracle.
- The Random Oracle model is an ideal model of the hash function that is not achievable in practice.
- In this model, we assume there exists an oracle h such that on input x $\in\{0,1\}^*$,if h has not seen x before, then it outputs a random value as h(x). Otherwise, it returns h(x) it previously output.
- Unfortunately, a random oracle does not exist since it requires infinite storage, so in practice we use pseudo-random functions.

Non-malleability

- Given h(x), it is hard to generate h(x’) where x an x’ are related

  exp: x’=x+1

Are these proporties imply others?

- Collision resistance (CR) -> Target collision resistance (TCR) (but not reverse)
- if h is OW is it also CR and TCR?
- if h is TCR or CR, is it also OW?

CRIPTOGTAFIC HASH FUNCTION APPLICATIONS

- Password Storage
- File Modificatition Detector
- Digital Signature
- Commitment

Password Storage

- We can store hash h(p) for password p instead of p directly, and check h(p) to authenticate a user.
- If it satisfies the property OW, adversary comprising h(p) will not learn p.
- What about CR and TCR?

File Modificatition Detector

- For each file F, we can store h(F) in a secure location. To check authenticity of a file, we can recompute h(F).
- What is a successfull break?
  - Adversary want to modify F to F’ but keep h(F)=h(F’)
- This requires property TCR.
- What about OW?

Digital Signature

- We can use hash functions to generate a signature that guarantees that the message came from a said source.
- Signing: ∂=sign(SK,M)
- Verificaiton : Verify(M, ∂, PK)= True/False

M, H(M) OW?

Commitment

- In a secure bidding, Alice wants to bid value x, but does not want to reveal the bid until the auction is over.
- Alice then computes C(x), and publicize it, which serves as her commitment.
- When bidding is over, then she can reveal x,and x can be verified using C(x).

Commitment

- C(x):
  - It must not reveal X (OW)
  - It must also protect system from Alice (CR)
  - Given C(X) should not be possible to produce C(X+1) (NM)