Hash Functions

Data and Information Management: ELEN 3015

School of Electrical and Information Engineering,
University of the Witwatersrand

Overview

1. Hash functions - Introduction

Hash is a one-way function $\rightarrow$ almost impossible to decrypt a hash into the original message

Hash function produces fixed size output, regardless of the size of input block

Encryption process which yields a fingerprint / signature

Finding hash is easy, finding message corresponding to hash is practically impossible

1. Hash functions - Introduction

Why should hash be unique?

1. Hash functions - Introduction

Why should hash be unique?

Alice signs M by $h = H(M)$

Mallory produces M' where $H(M) = H(M')$

Mallory can claim that Alice signed M', where M' favours Mallory and defrauds Alice

1. Hash functions - Introduction

Mathematically:

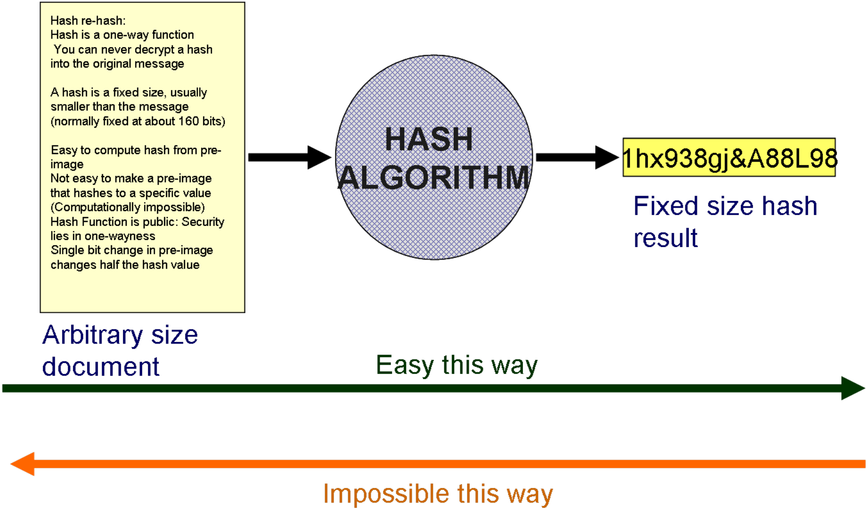
A. One-way hash function $H(M)$ operates on message M of any length, returns fixed length hash value h:

$$h = H(M)$$

B. Characteristics:

- Given M, computationally easy to compute h
- Given h, hard to compute arbitrary message M such that $H(M) = h$
- Given M, it is hard to find M' such that $H(M) = H(M')$

1. Hash functions - Introduction



2. Uses of hash functions

1. Passwords

Login	Password #
Bob	tu$jg
Alice	GG$$h3
James	xl5!$$

No need to store actual password, store only hash

2. Uses of hash functions

2. Signing documents

Hash function is unique to particular document $\rightarrow$ ‘fingerprint’

Cannot invent a document corresponding to a given hash

When computing hash of document $\rightarrow$ equivalent to signing document itself

Computationally cheaper to compute hash than public-key encrypt whole document

Authentication and integrity

2. Uses of hash functions

2. Signing documents

Hash re-hash:
Hash is a one-way function
You can never decrypt a hash into the original message

A hash is a fixed size, usually smaller than the message (normally fixed at about 160 bits)

Easy to compute hash from pre-image
Not easy to make a pre-image that hashes to a specific value (Computationally impossible)
Hash Function is public: Security lies in one-wayness
Single bit change in pre-image changes half the hash value

1hx938gj&A88L98

3. Length of hash

Hashes subject to "birthday attack" (birthday paradox)

Two approaches:

Naïve approach $\rightarrow$ Birthday paradox where someone tries to find another person with same birthday $\rightarrow$ number of documents created and hashed $= 2^{hashsize}/2$

Less naïve approach $\rightarrow$ Birthday paradox where someone tries to find any two people in a room with the same birthday $\rightarrow$ number of documents created and hashed $= 2^{hashsize/2}$

Due to birthday attack $\rightarrow$ hash length should be twice as long to secure against brute force attack

3. Length of hash

Naïve approach $\rightarrow$ Birthday paradox where someone tries to find another person with same birthday $\rightarrow$ number of documents created and hashed $= 2^{hashsize}/2$

Derive an equation for the probability $q(n)$ for the naive approach (for sharing a birthday).

Show that for the probability $q(n)$ to exceed 50 % we need $n = 253$.

3. Length of hash

Naïve approach $\rightarrow$ Birthday paradox where someone tries to find another person with same birthday $\rightarrow$ number of documents created and hashed $= 2^{hashsize}/2$

Derive an equation for the probability $q(n)$ for the naive approach (for sharing a birthday).

Show that for the probability $q(n)$ to exceed 50 % we need $n = 253$.

$$q(n) = 1 - \left(\frac{365 - 1}{365}\right)^n$$

3. Length of hash

Less naïve approach $\rightarrow$ Birthday paradox where someone tries to find any two people in a room with the same birthday $\rightarrow$ number of documents created and hashed $= 2^{hashsize/2}$

Derive an equation for the probability $p(n)$ for the less naive approach (any two persons sharing a birthday).

Show that for the probability $p(n)$ to exceed 50 % we need $n = 23$.

3. Length of hash

Less naïve approach $\rightarrow$ Birthday paradox where someone tries to find any two people in a room with the same birthday $\rightarrow$ number of documents created and hashed $= 2^{hashsize/2}$

Derive an equation for the probability $p(n)$ for the less naive approach (any two persons sharing a birthday).

Show that for the probability $p(n)$ to exceed 50 % we need $n = 23$.

$$p(n) = 1 - \left(\frac{365!}{365^n (365 - n)!} \right)$$

4. Hash Functions

MD5 $\rightarrow$ Discussed in notes $\Rightarrow$ Not for examination

SNERFU

N-HASH

MD4

MD2

etc.

Summary

Hash functions - Introduction

Uses of hash functions

Length of hash

Hash Functions