# Systems' Security | *Segurança de Sistemas*

## Cryptographic Hash Algorithms

Miguel Frade

POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

# Overview

# OVERVIEW

Learning Objectives

Introduction

Applications

Requirements

Authentication Algorithms

Exercises

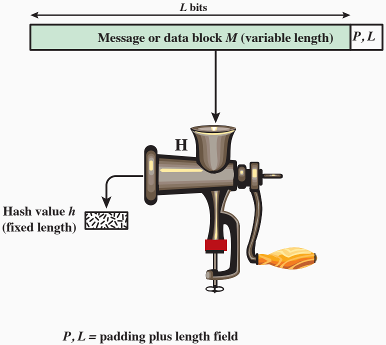
# Learning Objectives

After this chapter, you should be able to:

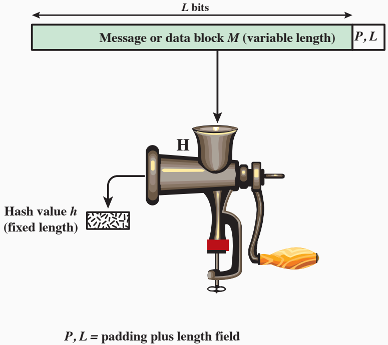
1. Summarize the applications of cryptographic hash functions
2. Explain why a hash function used for message authentication needs to be secured
3. Understand the differences among preimage resistant, second preimage resistant, and collision resistant properties
4. Present an overview of the basic structure of cryptographic hash functions
5. Understand the birthday paradox and present an overview of the birthday attack

# Introduction

Hash algorithms, or hash functions

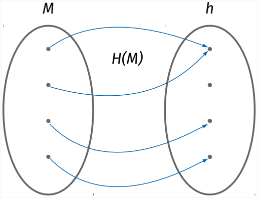


Hash algorithms, or hash functions

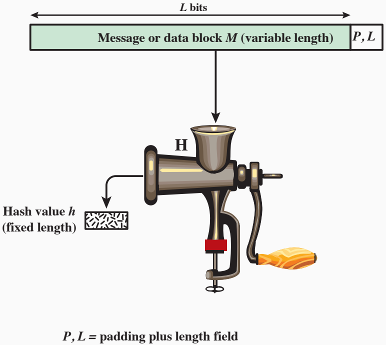



Many-to-one function

- input – a message $M$ of variable length
- output – a value $h$ of fixed length, *e. g.* 256 bits

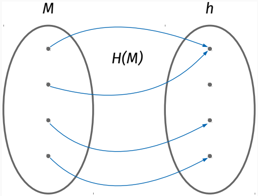


Hash algorithms, or hash functions




Many-to-one function

- input – a message $M$ of variable length
- output – a value $h$ of fixed length, *e. g.* 256 bits




- size of the messages $M$ universe $= \infty$
- size of the hash values $h$ universe $= 2^{n\ bits}$

## Definitions

- hash function
    - accepts a variable-length block of data $M$ as input and produces a fixed-size hash value $h = H(M)$
    - if applied to a large set of inputs the output should be evenly distributed and apparently random
    - a change to any bit or bits in $M$ results, with high probability, in a change to the hash value
    - are used to determine whether or not data has changed, that is, to verify data integrity

## Definitions

- hash function
  - accepts a variable-length block of data $M$ as input and produces a fixed-size hash value $h = H(M)$
  - if applied to a large set of inputs the output should be evenly distributed and apparently random
  - a change to any bit or bits in $M$ results, with high probability, in a change to the hash value
  - are used to determine whether or not data has changed, that is, to verify data integrity

**Cryptographic hash function**

It must be computationally infeasible to find either:

- a data object that maps to a pre-specified hash result (the one-way property)
- two data objects that map to the same hash result (the collision-free property)

# Applications

Cryptographic Hash Functions are very versatile and can be used for:

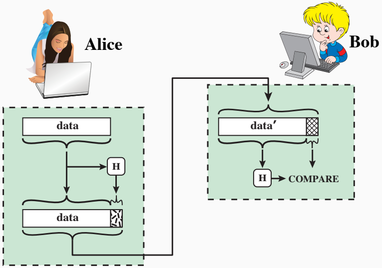
- data authentication
- digital signatures
- non-reversal password storage
- intrusion and virus detection
- pseudorandom number generator (PRNG)

## Data Authentication

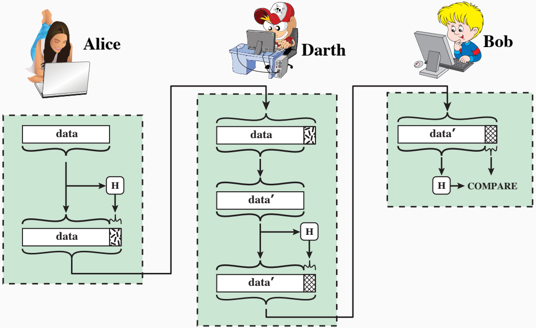
- to verify the integrity of a message assuring that data received are exactly as sent
  - there is no modification, insertion, deletion, or replay
- to assure the identity of the sender
- the hash value must be securely transmitted

## Data Authentication

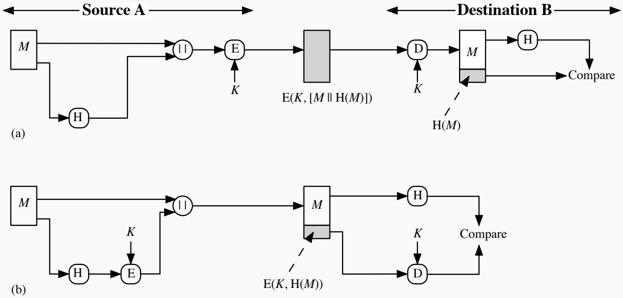
- to verify the integrity of a message assuring that data received are exactly as sent
    - there is no modification, insertion, deletion, or replay
- to assure the identity of the sender
- the hash value must be securely transmitted




$$A \rightarrow B : \quad M \parallel H(M)$$



**Data Authentication** – hash value protected with encryption



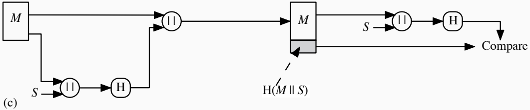

(a) $A \rightarrow B : E_k(M \parallel H(M))$

provides confidentiality of $M$

(b) $A \rightarrow B : M \parallel E_k(H(M))$

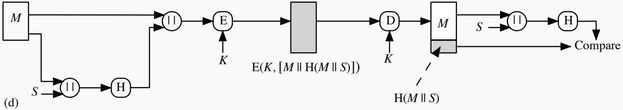
$M$ can be read by anyone

**Data Authentication** – hash value protected without encryption




(c) $A \rightarrow B : M \parallel H(M \parallel S)$

$M$ can be read by anyone



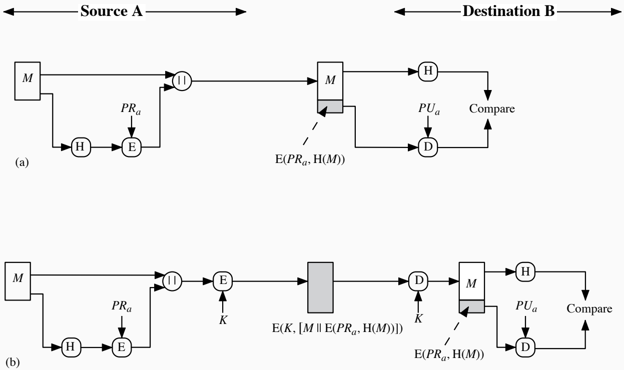

(d) $A \rightarrow B : E_k(M \parallel H(M \parallel S))$

provides confidentiality of $M$

**Data Authentication** – Message Authentication Code (MAC)

- other way to authenticate without encryption
- also know as keyed hash functions
- MAC will be addressed on another section

## Digital Signatures




(a) $A \rightarrow B : M \parallel E_{PR_A}(H(M))$

$M$ can be read by anyone

(b) $A \rightarrow B : E_k[M \parallel E_{PR_A}(H(M))]$

provides confidentiality of $M$

# Requirements

## Practical requirements

- Variable input size – $H$ can be applied to a block of data of any size

### Practical requirements

- Variable input size – $H$ can be applied to a block of data of any size
- Fixed output size – $H$ produces a fixed-length output

## Practical requirements

- Variable input size – $H$ can be applied to a block of data of any size
- Fixed output size – $H$ produces a fixed-length output
- Efficiency – $H(M)$ is relatively easy to compute for any given $M$, making both hardware and software implementations practical

## Definitions

- Preimage – For a hash value $h = H(M)$, we say that $M$ is the preimage of $h$

## Definitions

- Preimage – For a hash value $h = H(M)$, we say that $M$ is the preimage of $h$
- Collision
  - A collision occurs if we have $M \neq N$ and $H(M) = H(N)$
  - Collisions are undesirable for data integrity

## Cryptographic requirements

### Preimage resistant

For any given hash value $h$, it is computationally infeasible to find $N$ such that $H(N) = h$, *i. e.* the hash function is not reversible.

## Cryptographic requirements

### Preimage resistant

For any given hash value $h$, it is computationally infeasible to find $N$ such that $H(N) = h$, *i. e.* the hash function is not reversible.

### Second preimage resistant (weak collision resistant)

For any given block $M$, it is computationally infeasible to find $N \neq M$ with $H(N) = H(M)$.

## Cryptographic requirements

**Preimage resistant**

For any given hash value $h$, it is computationally infeasible to find $N$ such that $H(N) = h$, *i. e.* the hash function is not reversible.

**Second preimage resistant (weak collision resistant)**

For any given block $M$, it is computationally infeasible to find $N \neq M$ with $H(N) = H(M)$.

**Collision resistant (strong collision resistant)**

It is computationally infeasible to find any pair $(M, N)$ with $M \neq N$, such that $H(M) = H(N)$.

## Cryptographic requirements

**Preimage resistant**

For any given hash value $h$, it is computationally infeasible to find $N$ such that $H(N) = h$, *i. e.* the hash function is not reversible.

**Second preimage resistant (weak collision resistant)**

For any given block $M$, it is computationally infeasible to find $N \neq M$ with $H(N) = H(M)$.

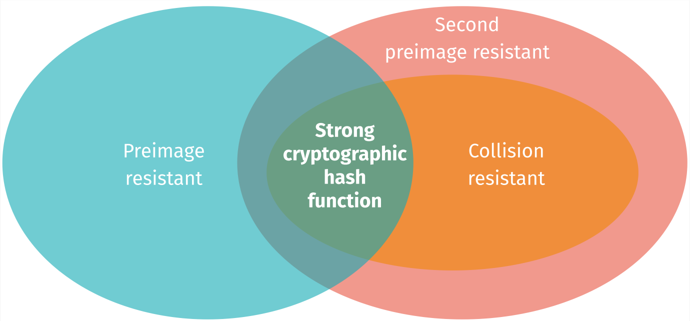
**Collision resistant (strong collision resistant)**

It is computationally infeasible to find any pair $(M, N)$ with $M \neq N$, such that $H(M) = H(N)$.

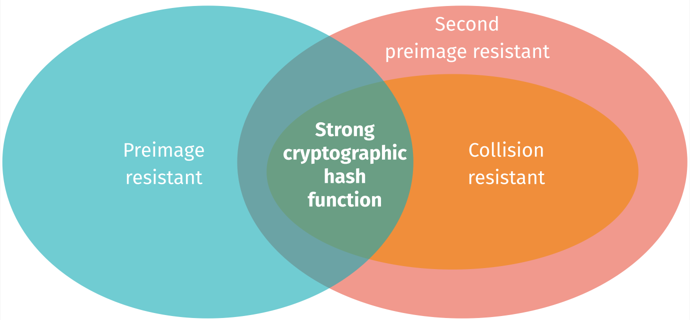
**Pseudorandomness**

Output of H meets standard tests for pseudorandomness.

Cryptographic requirements



## Cryptographic requirements




Effort to attack a hash of *m* bits

| Resistance | Operations |
|---|---|
| preimage | $2^m$ |
| $2^{nd}$ preimage | $2^m$ |
| collision | $2^{\frac{m}{2}}$ |

Cryptographic requirements for specific applications

| | Type of resistance | | |
|---|---|---|---|
| **Application** | **Preimage** | $2^{nd}$ **preimage** | **Collision** |
| digital signatures + hash | yes | yes | yes * |
| MAC | yes | yes | yes * |
| password storage | yes | — | — |
| IDS and virus | — | yes | — |
| encryption + hash | — | — | — |

* to protect against a chosen message attack

## Chosen message attack

As {the | --} Dean of Blakewell College, I have {had the pleasure of knowing | known} Cherise Rosetti for the {last | past} four years. She {has been | was} {a tremendous | an outstanding} {asset to | role model in} {our | the} school. I {would like to take this opportunity to | wholeheartedly} recommend Cherise for your {school's | --} graduate program. I {am | feel} {confident | certain} {that | --} {she | Cherise} will {continue to | --} succeed in her studies. {She | Cherise} is a dedicated student and {thus far her grades | her grades thus far} {have been | are} {exemplary | excellent}. In class, {she | Cherise} {has proven to be | has been} a take-charge {person | individual} {who is | --} able to successfully develop plans and implement them.

{She | Cherise} has also assisted {us | --} in our admissions office. {She | Cherise} has {successfully | --} demonstrated leadership ability by counseling new and prospective students. {Her | Cherise's} advice has been {a great | of considerable} help to these students, many of whom have {taken time to share | shared} their comments with me regarding her pleasant and {encouraging | reassuring} attitude. {For these reasons | It is for these reasons that} I {highly recommend | offer high recommendations for} Cherise {without reservation | unreservedly}. Her {ambition | drive} and {abilities | potential} will {truly | surely} be an {asset to | plus for} your {establishment | school}.

Letter with $2^{38}$ variations

# Authentication Algorithms

3. **Hash functions** main characteristics
    - are one-way functions → cannot be reversed
    - have fixed length output, regardless of the input size
    - <u>the hash value must be protected</u>

3. **Hash functions** main characteristics

- are one-way functions → cannot be reversed
- have fixed length output, regardless of the input size
- the hash value must be protected

Are used to:

- data integrity verification
- digital signatures schemes
- crypto-currency
- file control version (*e.g.* git)
- find duplicate files
- intrusion detection systems and anti-virus

3. **Hash functions** main characteristics

- are one-way functions $\rightarrow$ cannot be reversed
- have fixed length output, regardless of the input size
- the hash value must be protected

Are used to:

- data integrity verification
- digital signatures schemes
- crypto-currency
- file control version (*e.g.* git)
- find duplicate files
- intrusion detection systems and anti-virus
- cryptographic schemes
  - Message Authentication Codes (MAC), also know as keyed hash
  - Key Derivation Functions (KDF)
  - One-Time-Password (OTP)

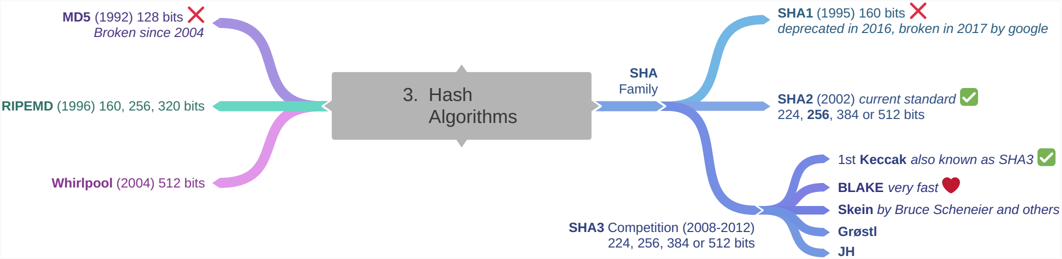
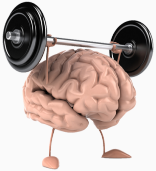

MD5 (1992) 128 bits
Broken since 2004
RIPEMD (1996) 160, 256, 320 bits
Whirlpool (2004) 512 bits
3. Hash Algorithms
SHA Family
SHA1 (1995) 160 bits
deprecated in 2016, broken in 2017 by google
SHA2 (2002) current standard
224, 256, 384 or 512 bits
SHA3 Competition (2008-2012)
224, 256, 384 or 512 bits
1st Keccak also known as SHA3
BLAKE very fast
Skein by Bruce Scheneier and others
Grøstl
JH

**Table 1:** Comparable strengths to resist a brute-force attack

| Bits | Symmetric | Hash | ECC | RSA/DH/DSA |
|---|---|---|---|---|
| 80 | 2DES | SHA1 (160) | 160 – 223 | 1 024 |
| 112 | 3DES | SHA2-224 | 224 – 255 | 2 048 |
| 128 | AES-128 | SHA2-256 | 256 – 383 | 3 072 |
| 192 | AES-192 | SHA2-384 | 384 – 511 | 7 680 |
| 256 | AES-256 | SHA2-512 | ⩾512 | 15 360 |

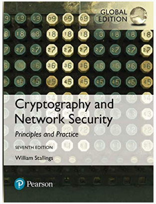
Source: NIST SP 800-57 Pt. 1 Rev. 4 (`https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-4/final`)

# Exercises



1. What are the 6 characteristics needed in a secure hash function?
2. What is the difference between weak and strong collision resistance?
3. Why it is required to protect the hash value?
4. In what ways can a hash value be secured so as to provide message authentication?

# Questions?




**Chapters 11** of

*William Stallings*, Cryptography and Network Security: Principles and Practice, Global Edition, 2016