ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

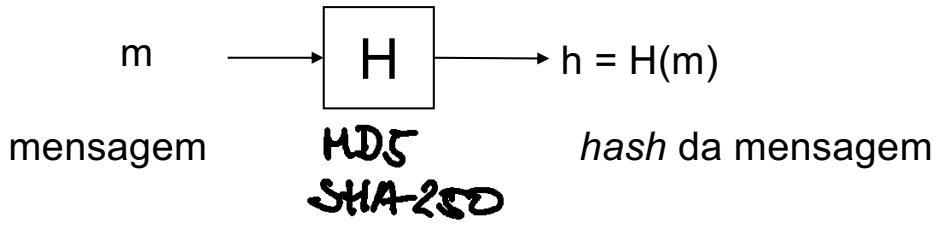
Funções de hash

ISEL – Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emídio Navarro, 1 | 1959-007 Lisboa

No segmento dos MAC

Funções de *hash*

- Função de *hash* criptográfica
  - **H: {0,1}* → {0,1}^{n}**, onde **n** é a dimensão do *hash*
  - Entrada:
    - Sequências binárias de dimensão finita
  - Saída:
    - Sequência binária de dimensão fixa (**n**)
  - **n** é a dimensão do *hash*



echo "this is a string" | openssl digest -sha384
echo "this is a string" | openssl digest --sha3-256

usar este

sha2 - é vistima de ataques

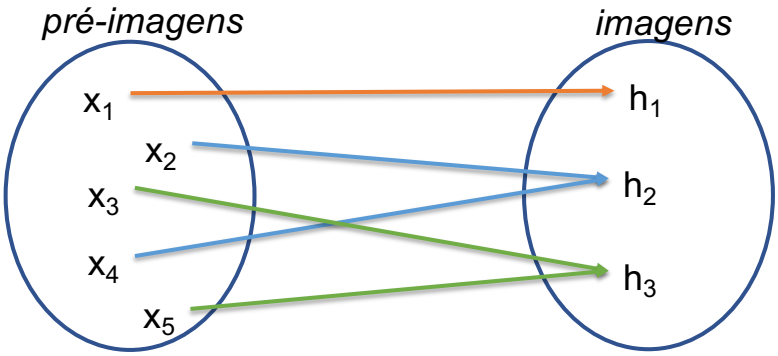
Notas

- Propriedades de segurança
  - **É computacionalmente fácil obter H(x) dado x** fácil de calcular
  - **É computacionalmente difícil, dado x, obter x'≠x tal que H(x') = H(x)**
    - Segunda pré-imagem não deve ser invertível
  - **É computacionalmente difícil obter (x, x'), com x'≠x, tal que H(x) = H(x')**
    - colisão Encontrar 2 pré imagens com o mesmo valor

- O hash de m serve como representante ("impressão digital") de m
- Exemplos de dimensões: MD5 (n=128) e SHA-1 (n=160)
- Baseiam-se em operações booleanas e aritméticas sobre palavras de pequena dimensão (16, 32, 64 bit)

Segunda pré-imagem e colisões

- Segunda pré-imagem - Dado x_2 e h_2, encontrar x_4 deve ser computacionalmente difícil
- Colisão - Encontrar (x_3, x_5) ou (x_2, x_4) deve ser computacionalmente difícil



- Considere um conjunto de pré-imagens {x_1, x_2, x_3, x_4, x_5} e imagens {h_1, h_2, h_3}
- Exemplo: a função SHA256 tem 2^{256} imagens possíveis

Ver exemplo de colisão de 2 imagens em MD5

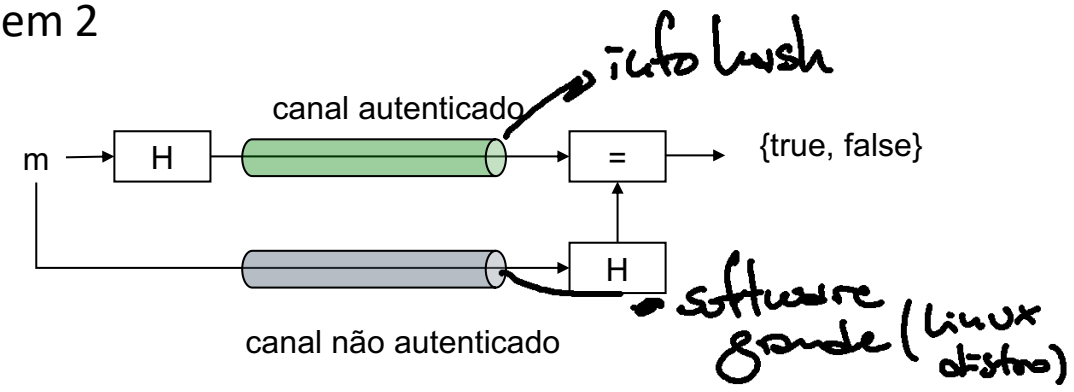
Exemplos de aplicação

- Garantia de integridade de dados
- Derivação de chaves a partir de *passwords* (Key Derivation Functions)
- Algoritmos de MAC
- Assinatura digital (esquema assimétrico)
- Diversos protocolos criptográficos
  - https://github.com/DP-3T/documents/blob/master/DP3T%20White%20Paper.pdf

↓ acompanhamento de contactos / COVID

Exemplo de utilização: integridade

- Exemplo: Distribuição de *software*
  1. Produtor calcula o *hash* da distribuição (ex. sources.tar.gz)
  2. Cliente obtêm, de forma autenticada, o *hash* da distribuição
  3. Cliente obtêm a distribuição (ex. através dum *mirror* não autenticado)
  4. Cliente compara o *hash* da distribuição recebida com o *hash* obtido em 2



Ex: https://downloads.apache.org/httpd/httpd-2.4.53.tar.bz2.sha512

Exemplo de comandos openssl e outros

```
$ openssl dgst -sha256 file.c
SHA256(file.c)= 0b2a06a29688...(omitted)...1f04ed41d1

$ openssl sha256 file.c
SHA256(file.c)= 0b2a06a29688...(omitted)...1f04ed41d1
```

```
$ md5sum file.c
919302e20d3885da126e06ca4cec8e8b   file.c

$ sha256sum file.c
0b2a06a29688...(omitted)...1f04ed41d1   file.c
```

Funções de *hash* com chave

- É usual designar-se um esquema de MAC, com algoritmo **T** determinístico, como *função de hash com chave* (*Keyed Hash Function*)
- HMAC é um conjunto de algoritmos MAC para usar com diferentes funções de hash H
- Exemplo da estrutura HMAC