# FUNZIONE DI HASH

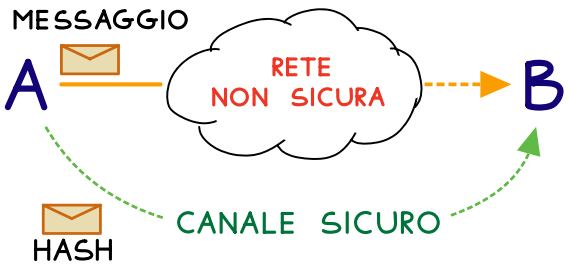
04/04/2023

Ci occupiamo ora di come garantire l'INTEGRITÀ, ovvero di come evitare che l'avversario possa intercettare e modificare i messaggi.

**FUNZIONE DI HASH:** Trasforma un messaggio $m$ di lunghezza variabile in un CODICE DI HASH $c$ di lunghezza fissa. $H(m) = c$

Tale funzione è essenziale in molte applicazioni poiché identifica delle informazioni con un codice.
Questo CODICE, come vedremo, può essere usato per fare delle verifiche di INTEGRITÀ, PROVENIENZA, CORRETTEZZA...

Una applicazione di utilizzo può essere la seguente:




Il mittente manda il messaggio attraverso una RETE NON SICURA.
Il destinatario lo riceve e me calcola HASH... e lo confronta con quello mandato dal mittente su una RETE SICURA.

**PROBLEMA:** COLLISIONI

Può capitare che due messaggi abbiamo lo stesso CODICE.
Questo accade poiché in media $|m| > |c|$

$$H(m_1) = c$$
$$H(m_2) = c$$

$$\langle m_1, m_2 \rangle = \text{COLLISIONE}$$

NEW PROBLEM

**PROBLEMA:** FALSIFICAZIONE

Le COLLISIONI rendono possibile la FALSIFICAZIONE di messaggi.

Vogliamo una FUNZIONE DI HASH che sia:

- **NON INVERTIBILE (ONE-WAY)**: Dato $c$, è difficile trovare $H^{-1}$.

  Poiché se è facile calcolare $H^{-1}$, sarà facile ottenere COLLISIONI.

  ↑ Include

- **FORTEMENTE NON INVERTIBILE**: Dato $m_1$, è difficile trovare $m_2$ $t.c.$ $H(m_1) = H(m_2)$

  ↑ Include

- **RESISTENTE ALLE COLLISIONI**: È difficile trovare $m_1, m_2$ $t.c.$ $H(m_1) = H(m_2)$

Per fare ciò dobbiamo analizzare il punto di vista del nostro avversario: come fa a generare COLLISIONI ?

Con un **ATTACCO DI FORZA BRUTA**: Guarda come è fatta la FUNZIONE DI HASH $H$ e capisce come trovare una COLLISIONE.

Questo tipo di attacco può essere facilitato di molto, basandosi sul **PARADOSSO DEL COMPLEANNO**.

Questo fenomeno è molto semplice e si basa sul fatto che è facile che due persone compiano gli anni lo stesso giorno, in una classe di $23^+$ persone.

$$P\left(\begin{array}{c}\text{nessun}\\ \text{compleanno}\\ \text{in comune}\end{array}\right) = \underbrace{(365 \cdot 364 \cdot \ldots \cdot (365-22))}_{\text{casi favorevoli}} / \underbrace{365^{23}}_{\text{casi possibili}} = 0,4927$$

$$P\left(\begin{array}{c}\text{compleanno}\\ \text{in comune}\end{array}\right) > \frac{1}{2}$$

Dietro c'è (ovviamente) un ragionamento matematico e può essere generalizzato:

$P(n,k) =$ probabilità che almeno una ripetizione in un insieme di $k$ elementi scelti tra $n$.

$$P(n,k) = 1 - (n \cdot (n-1) \cdot \ldots \cdot (n-k+1))/n^k >$$
$$> 1 - e^{-k\cdot(k-1)/2n} \approx 1 - e^{-k^2/2n}$$

$$\boxed{P(n,k) > \frac{1}{2} \quad \text{per} \quad k > 1,18 \cdot \sqrt{n}}$$

Nel caso del COMPLEANNO: $P(n,k) > \frac{1}{2}$ per $K > 22,54$.

## A che ci serve sapere questo?

Se ad esempio consideriamo un **CODICE HASH** di $128b$, l'intuizione ci dice che per generare una COLLISIONE, il mostro avversario debba generare $2^{128}$ messaggi.

Invece, grazie al PARADOSSO DEL COMPLEANNO sappiamo che ci vogliono $2^{64}$ messaggi.

## Possiamo generalizzare questo concetto:

$$P(n,k) = P(\text{COLLISIONE}) > 0,5 \quad \text{per} \quad k > 1,18 \cdot \sqrt{n} \approx \sqrt{n}$$

Nel mostro caso:
- $k =$ numero totale di MESSAGGI $= 2^m$
- $n =$ numero totale di CODICI HASH di $c$ bit $= 2^c$

$$P(\text{COLLISIONE}) > 0,5 \quad \text{per} \quad k > \sqrt{n} \Rightarrow 2^m > \sqrt{2^c} \Rightarrow 2^m > 2^{c/2} \Rightarrow m > c/2$$

## DIMOSTRAZIONE

$$P(n,k) = 1 - (n \cdot (n-1) \cdot \ldots \cdot (n-k+1))/n^k$$
$$> 1 - e^{-k\cdot(k-1)/2n} \approx 1 - e^{-k^2/2n}$$

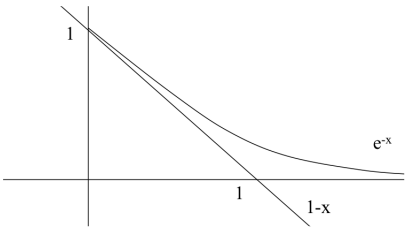
$$1 - (n \cdot (n-1) \cdot \ldots \cdot (n-k+1))/n^k =$$ ← divido per $n$ (cioè tolgo l'$n^k$ dal divisore e lo distribuisco)

$$1 - [1 \cdot (n-1)/n \cdot \ldots \cdot (n-k+1)/n] =$$ ← $\frac{n-1}{n} = \frac{n}{n} - \frac{1}{n}$

$$1 - [(1-1/n) \cdot \ldots \cdot (1-(k-1)/n)] >$$

$$1 - \left[\left(e^{-1/n}\right) \cdot \ldots \cdot \left(e^{-(k-1)/n}\right)\right] >$$

Uso il fatto che $e^{-x} > 1 - x$




---

$$P(n,k) > \frac{1}{2} \simeq 1 - e^{-k^2/2n} > \frac{1}{2}$$
$$-e^{-k^2/2n} > -\frac{1}{2}$$
$$e^{-k^2/2n} < \frac{1}{2}$$
$$e^{k^2/2n} > 2$$
$$k^2/2n > \ln(2)$$
$$k > \sqrt{2n \cdot \ln(2)}$$
$$k > 1,18 \cdot \sqrt{n}$$

Tornando a noi, come abbiamo detto, il **PROBLEMA** delle COLLISIONI può provocare la FALSIFICAZIONE. Questo accade quando il nostro avversario trova uno specifico messaggio con lo stesso **CODICE**.

Questo è quello che prende il nome di **PROBLEMA DEL COMPLEANNO** (BIRTDAY ATTACKS)

Leggermente diverso dal PARADOSSO DEL COMPLEANNO: una collisione qualunque!

$P'(n,k) =$ probabilità di trovare un elemento comune in due insiemi di $k$ elementi scelti tra $n$.

$$P'(n,k) = 1 - \left(\left(1 - \frac{1}{n}\right)^k\right)^k > 1 - \left(\left(e^{-1/n}\right)^k\right)^k = 1 - e^{-k^2/n}$$

$$\boxed{P'(n,k) > \frac{1}{2} \quad \text{per} \quad k > 0,83 \cdot \sqrt{n}}$$

Trovare la COLLISIONE però non basta... è necessario infatti avvalersi della collaborazione del mittente ...
→ cosa non sempre possibile praticamente (per fortuna).

Il PROBLEMA DEL COMPLEANNO è un esempio di **CHOSEN MESSAGGE ATTACK**
→ è l'avversario a scegliere il messaggio da cifrare.

**CONCLUSIONI:** Per garantire la sicurezza è innanzitutto fondamentale usare **CODICI DI HASH** grandi.

EXTRA: Altri esempi di tipologie di attacchi sono

**CIPHERTEXT ONLY ATTACK** → È di difficoltà massima poiché conosce solo il testo cifrato.

**KNOW MESSAGGE ATTACK** → È di difficoltà intermedia in quanto l'avversario conosce alcuni messaggi.

## DIMOSTRAZIONE

$$P'(n,k) = 1 - \left(\left(1 - \frac{1}{n}\right)^k\right)^k > 1 - \left(\left(e^{-1/n}\right)^k\right)^k = 1 - e^{-k^2/n}$$

$X = \{x_1, x_2, \ldots, x_k\}$
$Y = \{y_1, y_2, \ldots, y_k\}$

$$P(x_1 = y_1) = \frac{1}{n}$$

$$P(x_1 \neq y_1) = 1 - \frac{1}{n}$$

$$P((Y \cap \{x_1\}) = \phi) = \left(1 - \frac{1}{n}\right)^k$$

$$P((Y \cap \{x_1, x_2\}) = \phi = \left(1 - \frac{1}{n}\right)^{k^2}$$

$$P(Y \cap X) = \phi) = \left(\left(1 - \frac{1}{n}\right)^k\right)^k$$

come PARADOSSO DEL COMPLEANNO ...
uso il fatto che $e^{-x} > 1 - x$ ...

$$P'(n,k) > \frac{1}{2} \simeq 1 - e^{-k^2/n} > \frac{1}{2}$$

$$-e^{-k^2/n} > -\frac{1}{2}$$

$$e^{-k^2/n} < \frac{1}{2}$$

$$e^{k^2/n} > 2$$

$$k^2/n > \ln(2)$$

$$k > \sqrt{n \cdot \ln(2)}$$

$$k > 0,83 \cdot \sqrt{n}$$

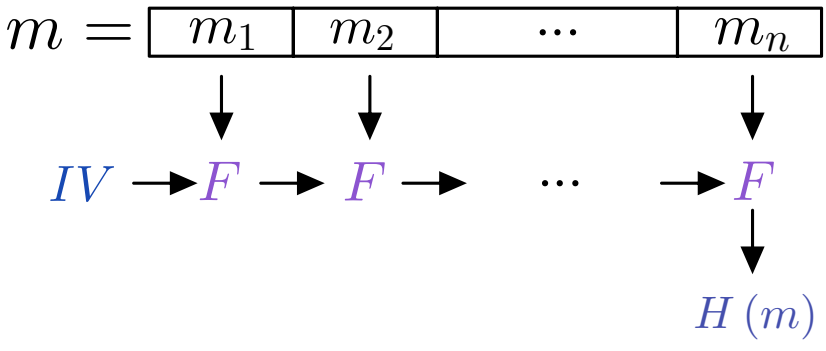
$$P(n,k) = P(\text{COLLISIONE}) > 0,5 \quad \text{per} \quad k > 0,83 \cdot \sqrt{n} \approx \sqrt{n}$$

Anche in questo caso quindi, se consideriamo un **CODICE** di $64b$, per generare una COLLISIONE ci vorranno $2^{32}$ messaggi.

# FUNZIONE DI HASH

**PREMESSE:** CODICI DI HASH sufficientemente LUNGHI!
RESISTENTE ALLE COLLISIONI

SCHEMA GENERALE:

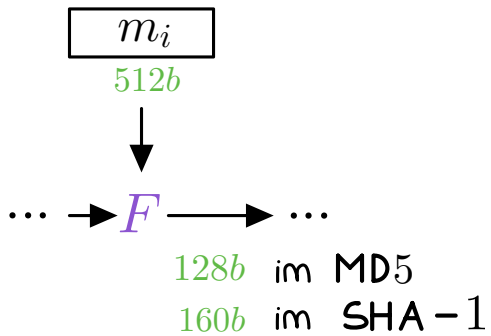



Divido il messaggio in $n$ parti, ciascuna di esse costituisce uno dei due input della sotto-funzione.

$F$ è la **FUNZIONE DI HASH** che, tra gli algoritmi più noti, può essere: **MD5** Message Digest 5
**SHA-1** Secure Hash Algorithm - v1

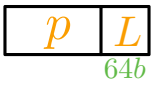
Non approfondiremo come sono fatte queste funzioni ... ma è importante sapere che sono molto complesse.

Nello SCHEMA di MD5/SHA-1:




Il messaggio viene diviso in blocchi di $512b$.

Nel caso sia messario si effettua una operazione di PADDING (riempimento):




$L \mod 2^{64}$
poiché altrimenti non ci starebbe nei $64b$

$p$: sequenza di bit riempiti

$L$: sono gli ultimi $64b$ ed indicano l'effettiva lunghezza del messaggio

Alla fine di tutto si ottiene il vero e proprio CODICE DI HASH anche detto DIGEST (riassunto)