

# Funzioni hash

Corso di Sicurezza  
a.a. 2019-20

**Alfredo De Santis**

Dipartimento di Informatica  
Università di Salerno

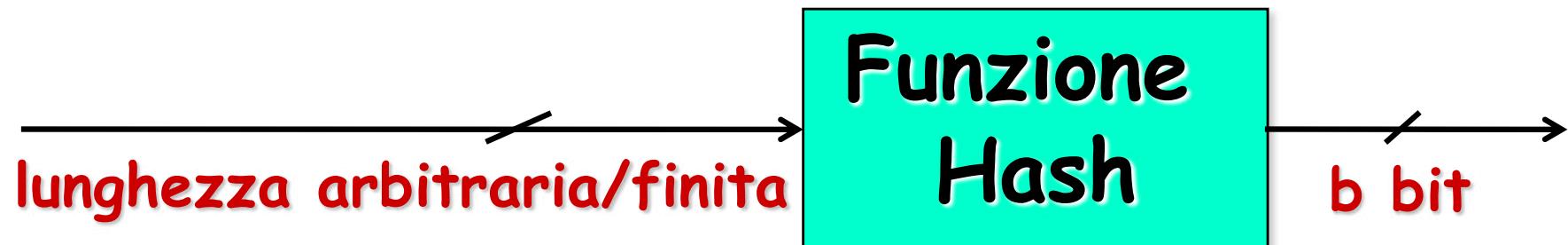
**ads@unisa.it**

**<http://www.di-srv.unisa.it/~ads>**



**Marzo 2020**

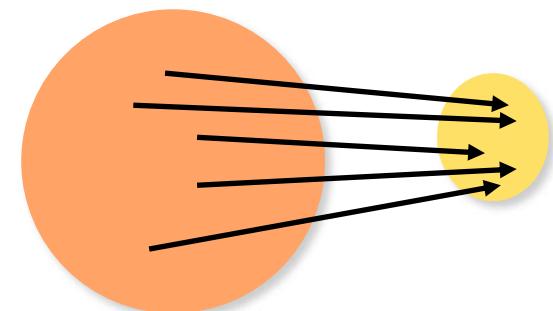
# Funzioni Hash



Idea alla base:

il valore hash  $h(M)$  è una rappresentazione non ambigua  
e non falsificabile del messaggio  $M$

Proprietà: comprime ed è  
facile da computare



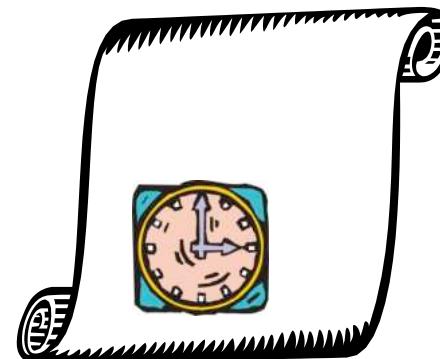
# Uso delle funzioni hash

Firme digitali



Integrità dei dati

Certificazione del tempo



# Firme digitali e Funzioni hash

**Problema:** firma digitale di messaggi lunghi

**Soluzione naïve:** Divisione in blocchi e firma per ogni blocco

problema per la sicurezza: una permutazione/composizione delle firme è una nuova firma

**Soluzione di uso corrente:**

firmare il valore hash del messaggio

$$[\text{firma di } M] = F_k(h(M))$$



**Vantaggi:** integrità dei dati ed efficienza degli algoritmi

# Integrità dei dati e Funzioni hash

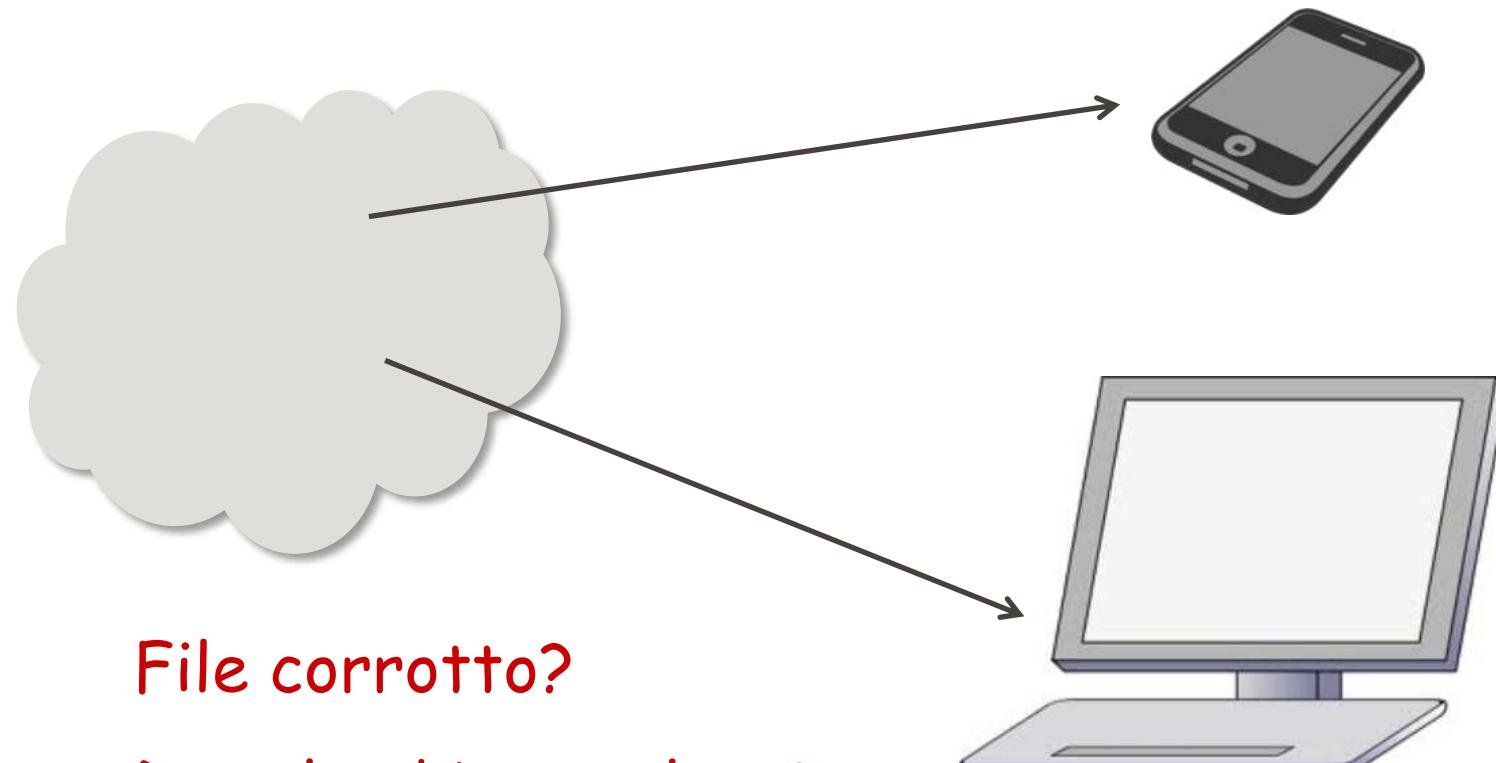
## Tipico uso delle funzioni hash

- Compuo al tempo T il valore hash del file M
- Conservo  $H = h(M)$  in un luogo sicuro
- Per controllare se il file è stato successivamente modificato, calcolo  $h(M')$  e verifico se  $H = h(M')$
- **$h(M)$  è l'impronta digitale del file**  
Assicura se un file è stato modificato!

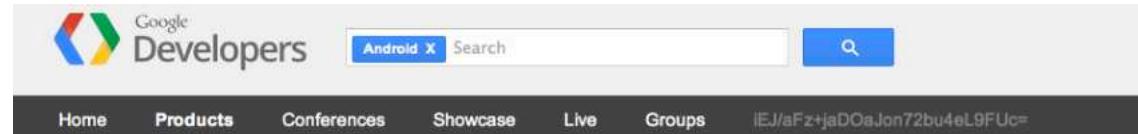


# Integrità dei dati e Funzioni hash

Download file



# Integrità dei dati e Funzioni hash



Home      Products      Conferences      Showcase      Live      Groups      IEJ/aFz+jaDOaJor72bu4eL9FUc=

Products > Android

Android 5.7k

---

Home      Nexus Binaries      **Nexus Factory Images**

## Factory Images for Nexus Devices

This page contains binary image files that are provided for use in restoring your Nexus device's original factory image. These files are intended for use only on your personal Nexus devices and may not be disassembled, decompiled, reverse engineered, or otherwise modified by you or used in any way except as specifically set forth in the license terms that came with your device.

### Factory Images "razorg" for Nexus 7 [2013] (Mobile)

Version	Download	MD5 Checksum	SHA-1 Checksum
4.3 (JLS36C)	<a href="#">Link</a>	186e8ac3b198276289a5ba3e1569a758	fb03a89feb7c60fb7e31e67c587da3c97c2bf56b
4.3.1 (JLS36I)	<a href="#">Link</a>	344feabad51d3bedb75a6e4d1d451a75	ecb320cdea2fa980d8a944dd82227bb8f89d58fd
4.4 (KRT16S)	<a href="#">Link</a>	20bee0445c67aa45c002a14e4bac1d77	bd6c92418035598c0815d19a2b21066d3f6fe9b6
4.4.2 (KOT49H)	<a href="#">Link</a>	8e42ffe324a9109031dd4f9dd53fdc9a	49789b240cab9a8e91c1f5a3eb0b6be0b07b5732
4.4.2_r2 (Verizon) (KVT49L)	<a href="#">Link</a>	2efd400254e657d0b72e698daff7ba4d	65bdb0e09288802998ac8834341fa2b7a0ecb9b7

### Factory Images "mantaray" for Nexus 10

Version	Download	MD5 Checksum	SHA-1 Checksum
4.2.2 (JDQ39)	<a href="#">Link</a>	b7a1162fb4e617143306ef6c4ca6c040	d79f489e1001d183b31d8a407b47cd5b8e9505cd
4.3 (JWR66Y)	<a href="#">Link</a>	60d0df743b44aee10f125be8e03e84f2	3d8252dd33af47ff7132734b053dffbc2b58ce4
4.4 (KRT16S)	<a href="#">Link</a>	7b308faf560cedd6970f27fc40828b2e	944139617036fca28848cf0ace67b81f93e08e4e
4.4.2 (KOT49H)	<a href="#">Link</a>	6812260ac97283bd0053e09a05cd5825	174ba74f19a22c0e96467287c34cb63c6e9f751d

aprile 2014

## Factory Images

[Full OTA Images](#)  
[Driver Binaries](#)

## "coral" for Pixel 4 XL

Version	Download	SHA-256 Checksum
10.0.0 (QD1A.190821.007, Oct 2019)	<a href="#">Link</a>	e915f51a4afdf2ec6b1a802e105a9e3427613e813d208ae38878a3973f5d0e6b6
10.0.0 (QD1A.190821.011, Oct 2019)	<a href="#">Link</a>	6dbd28f34a2db88e3c4c02104b7ed8ad1338cbe72585a5ac10d435a8a53a3e6c
10.0.0 (QD1A.190821.011.C4, Oct 2019)	<a href="#">Link</a>	6a29be3b076158de063248b1995fc275474d38a1178d67bb34a99df5a592db64
10.0.0 (QD1A.190821.007.A3, Nov 2019)	<a href="#">Link</a>	db77f854196b6a21d99e0cea6e4bee9e70a372423013226ecd7dbe64e1857110
10.0.0 (QD1A.190821.014, Nov 2019)	<a href="#">Link</a>	4ea7d6457f8c9edd59081d6281691522afe61ff3be3dd8b4e9fab893c6c1b5f5
10.0.0 (QD1A.190821.014.C2, Nov 2019)	<a href="#">Link</a>	839d41c9b1b733e9fec112915b376d7f2916a7f4a27625411d7dd0f518780ee7
10.0.0 (QQ1B.191205.011, Dec 2019, EMEA carriers, T-Mobile (US), Google Fi)	<a href="#">Link</a>	0c24f5e0eade8f038dba4a7f6a086575589143f940d0a29dfe06b2af3c4c3cf1
10.0.0 (QQ1B.191205.012.A1, Dec 2019)	<a href="#">Link</a>	a2b67bf7b2dbbb516fcac9f890041cf8aa362ee526eabdb783809635b82abbdb
10.0.0 (QQ1C.191205.016.A1, Dec 2019, Select JP & TW carriers)	<a href="#">Link</a>	fb333d0f15ea6d088bdf2cd0e9d1371fc5bcfc7d97f9b6ef03868cec4b056f3
10.0.0 (QQ1B.200105.004, Jan 2020)	<a href="#">Link</a>	143dfd87bda8e3064f041c1a97b3d4e3a34d0ac79977c79a7829c7df1532edaa
10.0.0 (QQ1C.200105.004, Jan 2020, Select JP & TW carriers)	<a href="#">Link</a>	7dcbe7c415184078308b173a7d4e03f442a214625e62d2d18faa663dfd802348
10.0.0 (QQ1D.200105.002, Jan 2020, NTT DOCOMO)	<a href="#">Link</a>	1e2dd846042f15acf0344b85405750bef08ff7d6d7fbef45e85400d7bb115c4
10.0.0 (QQ1B.200205.002, Feb 2020)	<a href="#">Link</a>	89d8438423ef5398e19bca74e1cc1088900ae2476e208b3b4be2860f8631f732
10.0.0 (QQ1C.200205.002, Feb 2020, Select JP & TW carriers)	<a href="#">Link</a>	2949c30167a4bbb8afe7f5cb291e1acc22211851b3b2ba5e69673044529e3449

maggio 2020

<https://developers.google.com/android/images>



## Apache OpenOffice 4.1.7 released

[home](#) » [download](#) » [checksums](#)

| [Product](#) | [Download](#) | [Support](#) | [Blog](#)

### Apache OpenOffice - Download checksum files

[Full installation sets](#)

[Language packs](#)

[Software Development Kit \(SDK\)](#)

[Source code](#)

[How to verify your download with ASC, MD5, SHA256 checksums?](#)

[Important Notes](#)

---

### Apache OpenOffice - Tested and released full installation sets

Language The names do not refer to countries.		Windows Intel EXE	Linux Intel RPM	Linux Intel DEB	Linux x86-64 RPM	Linux x86-64 DEB	Mac OS Intel DMG
Arabic	عربی	<a href="#">ASC MD5 SHA256</a>					
Asturian	Asturianu	<a href="#">ASC MD5 SHA256</a>					
Basque	Euskara	<a href="#">ASC MD5 SHA256</a>					
Chinese (simplified)	简体中文	<a href="#">ASC MD5 SHA256</a>					
Chinese (traditional)	正體中文	<a href="#">ASC MD5 SHA256</a>					
Czech	českina	<a href="#">ASC MD5 SHA256</a>					

814675f921dda6d42161797aeeef700b28706dc36961106af64ca8bb9209d2f34 \*Apache\_OpenOffice\_incubating\_3.4.1\_Win\_x86\_install\_zh-TW.exe

maggio 2020

[https://www.openoffice.org/download/checksums/3.4.1\\_checksums.html](https://www.openoffice.org/download/checksums/3.4.1_checksums.html)

## DOWNLOAD

### Scegliere l'edizione giusta

#### Verificare la tua immagine ISO

Scaricare il checksum SHA256 fornito da Linux Mint

Controllo di integrità

#### Controllo di autenticità

Importa la chiave di Linux Mint:

Verifica l'autenticità di  
sha256sum.txt:

## AVVIO IN MODALITÀ LIVE

### Creare il dispositivo di avvio

### Avviare Linux Mint

## INSTALLAZIONE

### Installare Linux Mint

## DOPO L'INSTALLAZIONE

### Driver hardware

### Codec multimediali

### Supporto lingue

### Istantanee (snapshot) del sistema

## RISOLUZIONE DEI PROBLEMI

### EFI

### Opzioni di avvio

## DOMANDE FREQUENTI

### Multi-boot

### Partizionamento

### Pre-installare Linux Mint (installazione OEM)

# Verificare la tua immagine ISO

È importante verificare l'integrità e l'autenticità della tua immagine ISO.

Il controllo di integrità conferma che la tua immagine ISO è stata scaricata in maniera corretta, e che il file locale che possiedi è una esatta copia del file presente nei server di download. Un errore durante lo scaricamento potrebbe risultare in un file corrotto, e potrebbe generare errori casuali durante l'installazione.

Il controllo di autenticità conferma che l'immagine ISO che hai scaricato è stata firmata da Linux Mint, e di conseguenza non è una copia modificata da qualcuno o contenente codice malevolo.

## Scaricare il checksum SHA256 fornito da Linux Mint

Tutti i [server per il download](#) forniscono le immagini ISO, un file [sha256sum.txt](#) ed un file [sha256sum.txt.gpg](#). Dovresti essere in grado di trovare quei file nello stesso posto da cui hai scaricato l'immagine ISO.

Se non riesci a trovarli, sfoglia il [server di download Heanet](#) e clicca sulla versione e l'edizione di Linux Mint che hai scaricato.

Scarica entrambi i file [sha256sum.txt](#) e [sha256sum.txt.gpg](#).

Do not copy their content, use «right-click->Save Link As...» to download the files themselves and do not modify them in any way.

## Controllo di integrità

Per verificare l'integrità del tuo file ISO locale, genera il suo checksum SHA256 e confrontalo con quello presente nel file [sha256sum.txt](#).

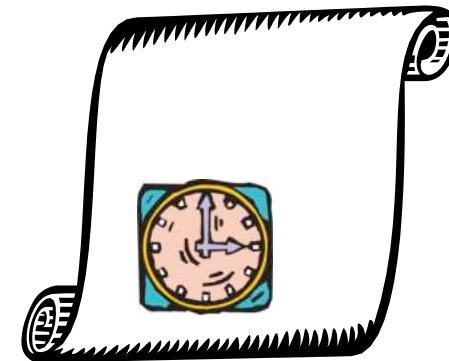
```
sha256sum -b yourfile.iso
```

# Certificazione del tempo e Funzioni Hash

Il notaio digitale

Quando è stato creato

il documento D ?



# Funzioni Hash: Proprietà

Facili da calcolare  
... poi?



# Un possibile attacco



prepara 2 versioni di un contratto  $M$  ed  $M'$

- $M$  è favorevole ad Alice
- $M'$  è sfavorevole ad Alice



modifica  $M'$  a caso (piccoli cambiamenti come aggiunta spazi: 32 possibilità sono  $2^{32}$  messaggi!) ) finchè  $h(M) = h(M')$

Alice firma  $M$   $\longrightarrow$   $\text{Firma}_{k\text{priv}}(h(M))$



ha quindi la firma di  $M'$   $\longrightarrow$   $\text{Firma}_{k\text{priv}}(h(M'))$

# Esempio di lettera fraudolenta

Cara Alice,

ti  $\left\{ \begin{array}{l} \text{scrivo} \\ \text{sto scrivendo} \end{array} \right\}$  da  $\left\{ \begin{array}{l} \text{un bellissimo} \\ \text{uno spendido} \end{array} \right\}$  posto  $\left\{ \begin{array}{l} \text{della costiera Amalfitana} \\ \text{vicino Amalfi} \end{array} \right\}$

.....

$\left\{ \begin{array}{l} \text{Colui} \\ \text{La persona} \end{array} \right\}$  che  $\left\{ \begin{array}{l} \text{ti porterà} \\ \text{è portatore di} \end{array} \right\}$  questa  $\left\{ \begin{array}{l} \text{lettera} \\ \text{missiva} \end{array} \right\}$  è di fiducia!

.....

2<sup>6</sup> varianti

# Funzioni hash: sicurezza

- **Sicurezza debole**: dato  $M$  è computazionalmente difficile trovare un altro  $M'$  tale che  $h(M) = h(M')$
- **Sicurezza forte**: computazionalmente difficile trovare 2 diversi messaggi con lo stesso valore hash
- **One-way**: dato  $y$  è computazionalmente difficile trovare  $M$  tale che  $y = h(M)$

# Sicurezza forte $\Rightarrow$ One-way

- $h: X \rightarrow Z$  funzione hash,  $|X| \geq 2 \cdot |Z|$ 
  - $\log |X| \geq \log (2 \cdot |Z|) = \log 2 + \log |Z| = 1 + \log |Z|$
  - Comprime di almeno 1 bit
- Supponiamo che **ALG** sia un algoritmo di inversione per  $h$
- ... allora esiste un algoritmo *Las Vegas* che trova collisioni con probabilità  $\geq 1/2$

# Sicurezza forte $\Rightarrow$ One-way

- $h: X \rightarrow Z$  funzione hash,  $|X| \geq 2 \cdot |Z|$
- Supponiamo che **ALG** sia un algoritmo di inversione per  $h$
- ... allora esiste un algoritmo *Las Vegas* che trova collisioni con probabilità  $\geq 1/2$

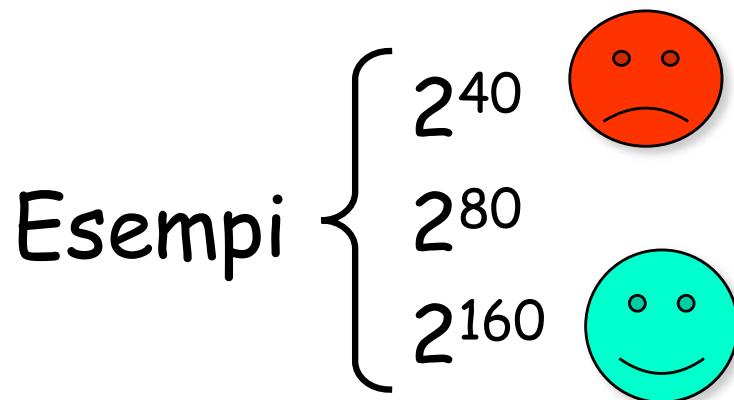
1. Scegli a caso  $x$  in  $X$
2.  $z \leftarrow h(x)$
3.  $x' \leftarrow \text{ALG}(z)$
4. If  $x' \neq x$  then  $(x', x)$  è una collisione  
else fallito

# Lunghezza valore hash

- Quanto grande l'hash per la sicurezza forte?
- Se  $|Z| = 2^3$  bit...
  - Quanti valori scegliere per essere certi di trovare almeno una collisione?

# Lunghezza valore hash

- Quanto grande l'hash per la sicurezza forte?
- Se  $|Z| = 2^3$  bit...
  - Quanti valori scegliere per essere certi di trovare almeno una collisione?
- $|Z|+1$  diversi valori di  $M$   
⇒ certezza di trovare almeno una collisione



# Lunghezza valore hash

- Nuovo attacco per trovare collisioni
  - $h: X \rightarrow Z$  funzione hash,  $|X| = m$  e  $|Z| = n$
  - Scelgo a caso diversi messaggi in  $X$
  - Verifico se ottengo almeno due valori hash uguali
- Quanti messaggi per avere una buona probabilità di successo?
  - $n$  numero dei diversi valori hash
  - $t$  numero messaggi da scegliere in  $X$
  - $\varepsilon$  probabilità di successo

# Paradosso del compleanno

Quante persone scegliere a caso affinchè, con probabilità  $\geq 0.5$ , ci siano almeno due con lo stesso compleanno?



# Paradosso del compleanno

Quante persone scegliere a caso affinchè, con probabilità  $\geq 0.5$ , ci siano almeno due con lo stesso compleanno?



*Risposta:* bastano 23 persone!

# Paradosso del compleanno

Probabilità che tra  $t$  valori scelti a caso ed indipendentemente non ci siano valori uguali

$$\frac{365 \cdot 364 \cdot \dots \cdot (365 - t + 1)}{365^t}$$

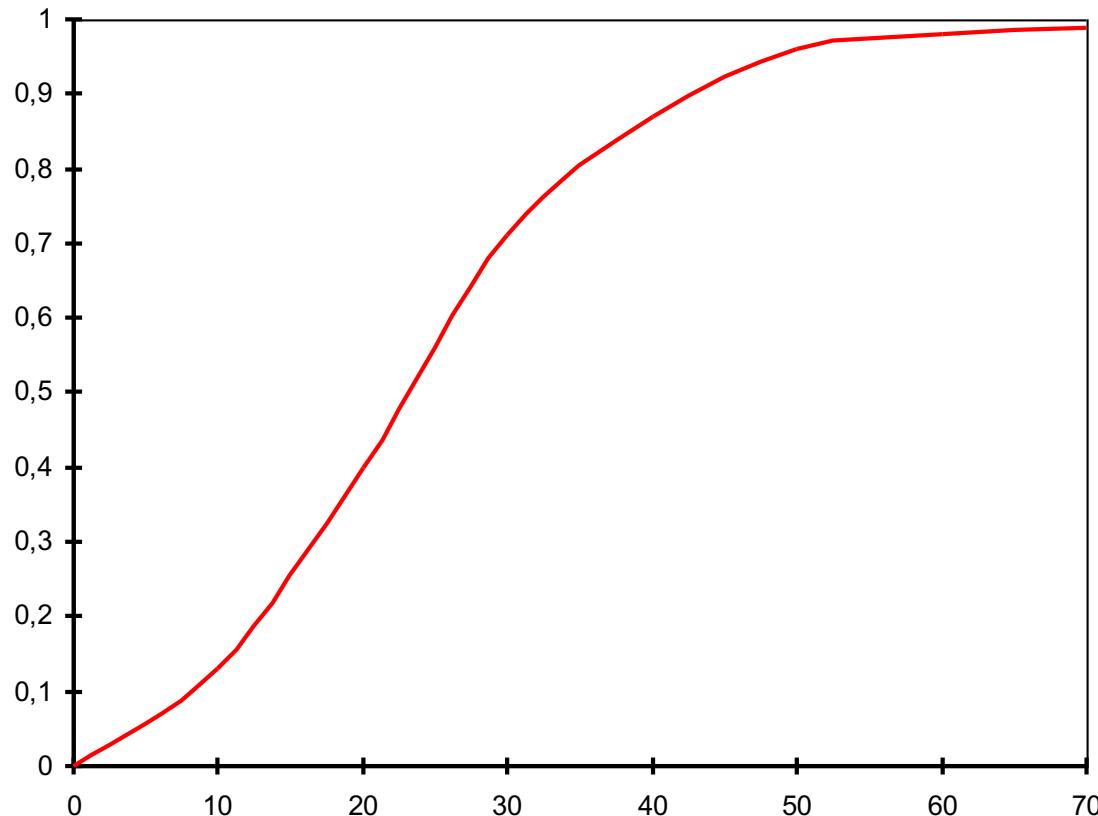
← casi favorevoli  
← totale casi

Probabilità che tra  $t$  valori scelti a caso ed indipendentemente almeno 2 sono uguali

$$1 - \frac{365 \cdot 364 \cdot \dots \cdot (365 - t + 1)}{365^t}$$

# Paradosso del compleanno

Grafico di  $1 - \frac{365 \cdot 364 \cdot \dots \cdot (365 - t + 1)}{365^t}$



$$t = 23 \rightarrow \varepsilon = 0.5073$$

$$t = 100 \rightarrow \varepsilon = 0.9999997$$

# Paradosso del compleanno

- Scegliamo a caso elementi in un insieme di cardinalità  $n$
- Quanti elementi scegliere se si vuole che la probabilità che ci siano almeno due elementi uguali sia  $\varepsilon$ ?

$$t \approx \sqrt{n \cdot 2 \ln\left(\frac{1}{1-\varepsilon}\right)}$$



# Paradosso del compleanno

- Scegliamo a caso elementi in un insieme di cardinalità  $n$ .
- Quanti elementi scegliere se si vuole che la probabilità che ci siano almeno due elementi uguali sia  $\varepsilon$ ?

$$t \approx \sqrt{n \cdot 2 \ln\left(\frac{1}{1-\varepsilon}\right)}$$

- Se  $\varepsilon = 0.5$  allora  $t \approx 1.17\sqrt{n}$
- Applicazione:  $n = 365$  e  $\varepsilon = 0.5$  allora  $t = 22.3$

# Paradosso del compleanno

- Scegliamo a caso elementi in un insieme di cardinalità  $n$ .
- Quanti elementi scegliere se si vuole che la probabilità che ci siano almeno due elementi uguali sia  $\varepsilon$ ?

$$t \approx \sqrt{n \cdot 2 \ln\left(\frac{1}{1-\varepsilon}\right)}$$

- Se  $\varepsilon = 0.5$  allora  $t \approx 1.17\sqrt{n}$
- Applicazione:  $n = 365$  e  $\varepsilon = 0.5$  allora  $t = 22.3$

Che relazione c'è con le funzioni hash?

# Attacco del compleanno

- Scegliere  $t$  elementi a caso e calcolarne i valori hash.
- Quanti elementi scegliere per avere almeno una collisione?
- Per una fissata probabilità  $\varepsilon$ ,  $t$  è circa  $\sqrt{n}$

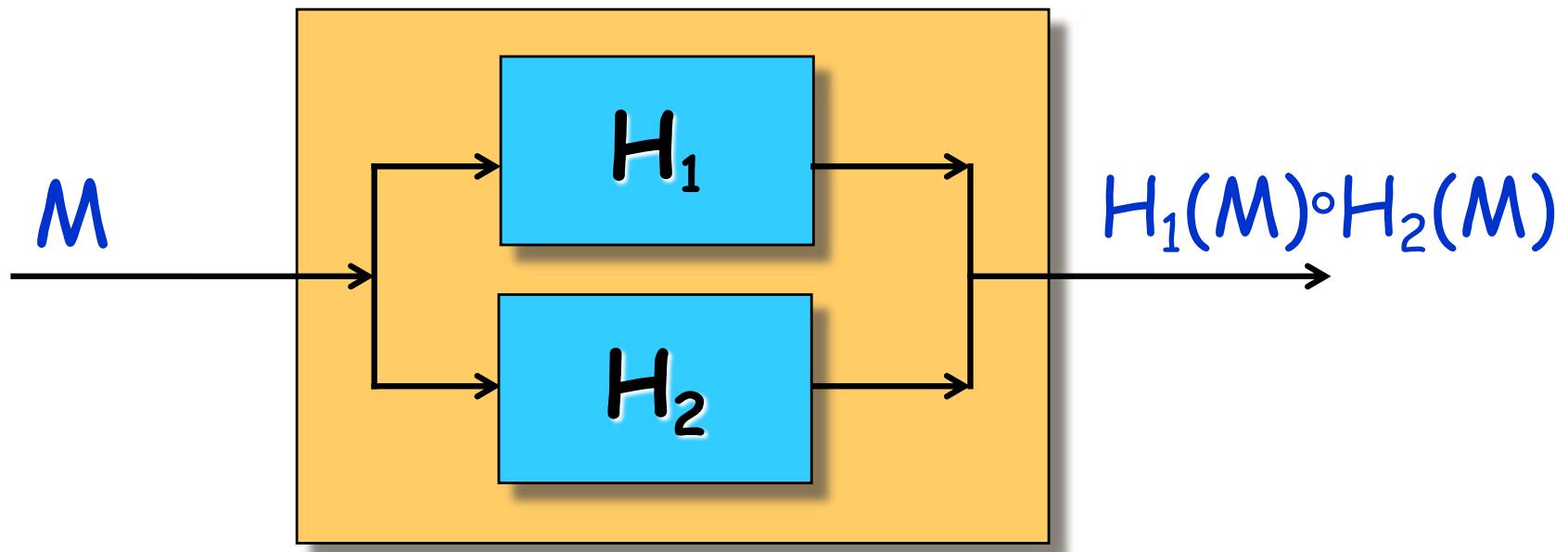
➤ Se  $n = 2^{80}$  allora  $t \approx 2^{40}$



➤ Se  $n = 2^{160}$  allora  $t \approx 2^{80}$



# Composizione funzioni hash



Trovare una collisione per  $H(M) = H_1(M) \circ H_2(M)$

→ trovare una collisione sia per  $H_1$  che per  $H_2$

# Modello generale per funzioni hash iterate

Input taglia arbitraria  $\rightarrow$  taglia fissata

Input M. Padding ed aggiunta della lunghezza di M.  
Si ottiene un messaggio con blocchi di taglia uguale  $X_1X_2\dots X_n$



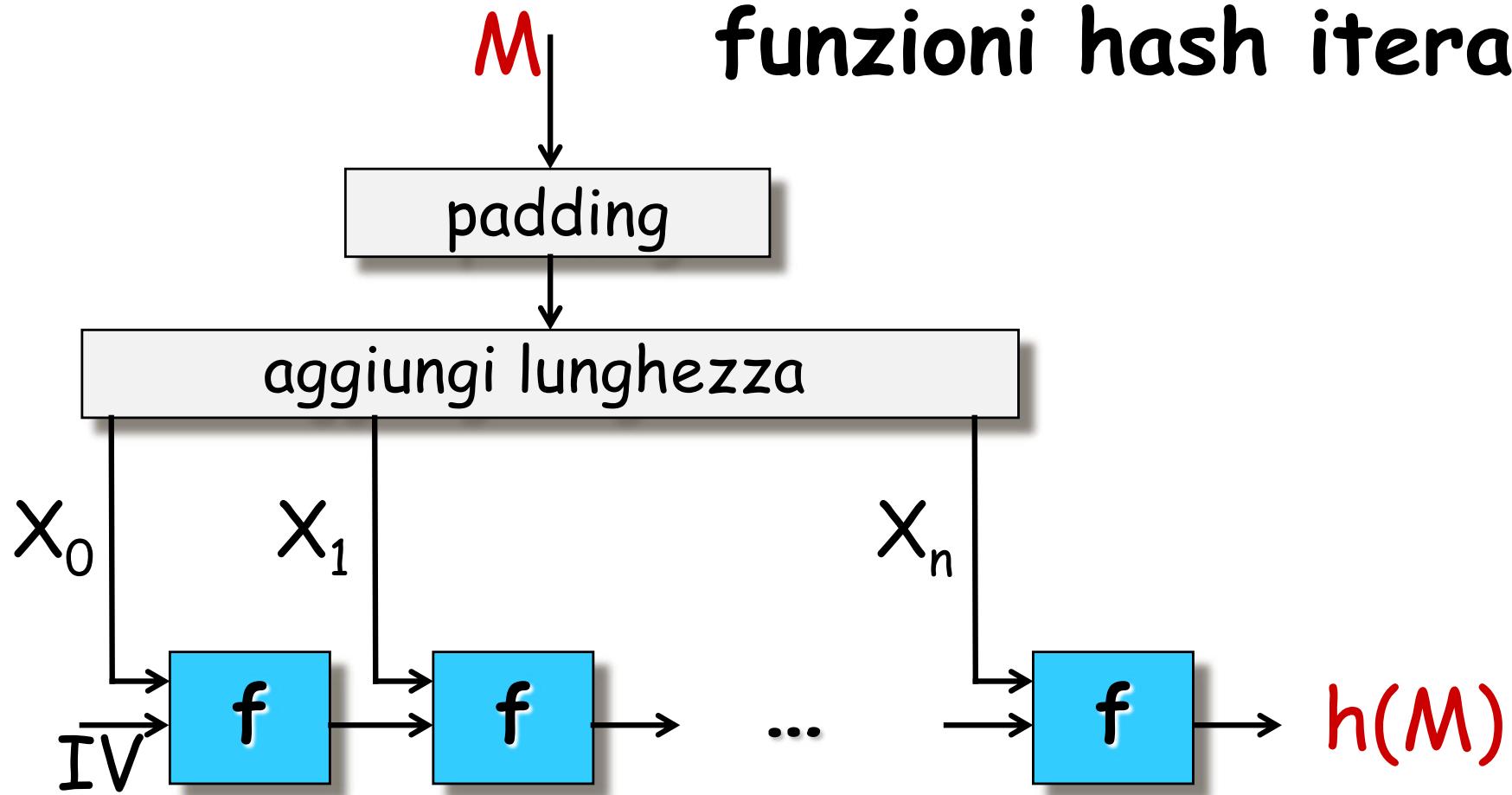
$H_0$  è una costante iniziale

Computazione di ...  $H_i = f(X_i, H_{i-1}) \dots$

Valore hash  $H_n = f(X_n, H_{n-1})$

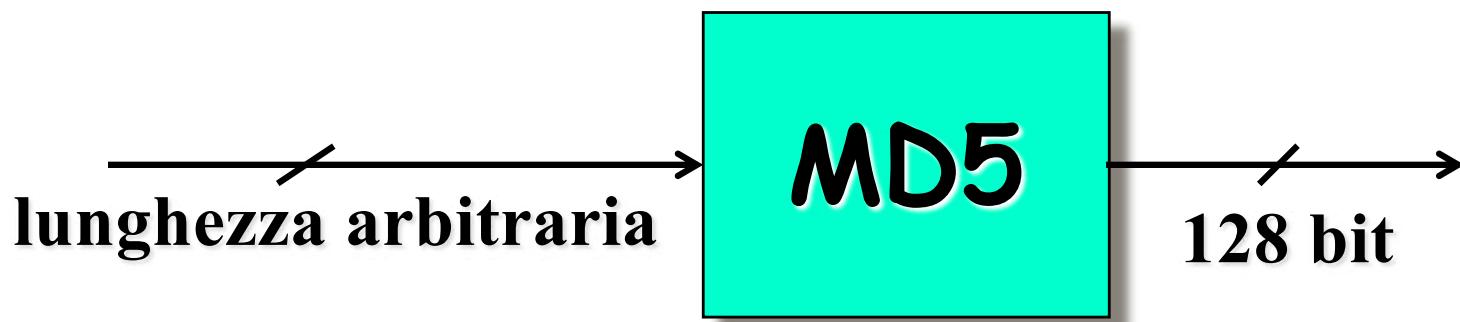
**computazione  
del valore hash**

# Modello generale funzioni hash iterate



# MD5

- Progettato nel 1991 da Ron Rivest
- MD da Message Digest
- Operazioni efficienti su architetture 32 bit little-endian
  - $a_1a_2a_3a_4$  rappresenta l'intero  $a_42^{24}+a_32^{16}+a_22^8+a_1$



# MD5: padding del messaggio

- MD5 processa il messaggio in blocchi di 512 bit
  - Ogni blocco consta di 16 parole di 32 bit
- M messaggio originario di b bit → padding

$$M' = \boxed{M \ 100\ldots0 \ b}$$

(447-b) mod 512 bit    64 bit

- $M'$  consta di un numero di bit multiplo di 512
  - ovvero di un numero di parole  $N$  multiplo di 16
  - quindi  $N/16$  blocchi di 512 bit

# MD5: operazioni

Funzioni definite su parole di 32 bit:

- round 1:  $F(X,Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$  (if  $X$  then  $Y$  else  $Z$ )  
round 2:  $G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$  (if  $Z$  then  $X$  else  $Y$ )  
round 3:  $H(X,Y,Z) = X \oplus Y \oplus Z$  (bit di parità)  
round 4:  $I(X,Y,Z) = Y \oplus (X \vee (\neg Z))$  (nuova funzione)

<b>X</b>	<b>Y</b>	<b>Z</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

# MD5: operazioni

Funzioni definite su parole di 32 bit:

- |   |                             |
|---|-----------------------------|
| round 1: $F(X,Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$ | (if $X$ then $Y$ else $Z$ ) |
| round 2: $G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$ | (if $Z$ then $X$ else $Y$ ) |
| round 3: $H(X,Y,Z) = X \oplus Y \oplus Z$                   | (bit di parità)             |
| round 4: $I(X,Y,Z) = Y \oplus (X \vee (\neg Z))$            | (nuova funzione)            |

X	Y	Z	F	G	H	I
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

Ogni round consiste di 16 operazioni [ABCD.k.s.i]

$$A \leftarrow B + ((A + W(B,C,D) + X[k] + T[i])) \ll s$$

- k è l'indice della parola, s indica lo shift ciclico, i è l'indice dell'iterazione, W è la funzione del round

# Costanti T[1 .. 64]

$T[i] \leftarrow \text{primi 32 bit di } |\sin(i)| = \lfloor 2^{32} \cdot |\sin(i)| \rfloor$  (i in radianti)

1	d76aa478	17	f61e2562	33	ffffa3942	49	f4292244
2	e8c7b756	18	c040b340	34	8771f681	50	432aff97
3	242070db	19	265e5a51	35	6d9d6122	51	ab9423a7
4	c1bdceee	20	e9b6c7aa	36	fde5380c	52	fc93a039
5	f57c0faf	21	d62f105d	37	a4beea44	53	655b59c3
6	4787c62a	22	02441453	38	4bdecfa9	54	8f0ccc92
7	a8304613	23	d8a1e681	39	f6bb4b60	55	ffeff47d
8	fd469501	24	e7d3fb8	40	bebfb70	56	85845dd1
9	698098d8	25	21e1cde6	41	289b7ec6	57	6fa87e4f
10	8b44f7af	26	c33707d6	42	ea127fa	58	fe2ce6e0
11	fffff5bb1	27	f4d50d87	43	d4ef3085	59	a3014314
12	895cd7be	28	455a14ed	44	04881d05	60	4e0811a1
13	6b901122	29	a9e3e905	45	d9d4d039	61	f7537e82
14	fd987193	30	fcefa3f8	46	e6db99e5	62	bd3af235
15	a679438e	31	676f02d9	47	1fa27cf8	63	2ad7d2bb
16	49b40821	32	8d2a4c8a	48	c4ac5665	64	eb86d391

# MD5

$A \leftarrow 0123456; B \leftarrow 89abcdef; C \leftarrow fdecba98; D \leftarrow 76543210;$

**for**  $i = 0$  **to**  $N/16-1$  **do**

**for**  $j = 0$  **to** 15 **do**  $X[j] \leftarrow M[16i+j]$

$AA \leftarrow A; BB \leftarrow B; CC \leftarrow C; DD \leftarrow D;$

[ABCD. 0.7. 1] [DABC. 1.12. 2] [CDAB. 2.17. 3] [BCDA. 3.22. 4]

[ABCD. 4.7. 5] [DABC. 5.12. 6] [CDAB. 6.17. 7] [BCDA. 7.22. 8]

[ABCD. 8.7. 9] [DABC. 9.12.10] [CDAB.10.17.11] [BCDA.11.22.12]

[ABCD.12.7.13] [DABC.13.12.14] [CDAB.14.17.15] [BCDA.15.22.16]

[ABCD. 1.5.17] [DABC. 6. 9.18] [CDAB.11.14.19] [BCDA. 0.20.20]

[ABCD. 5.5.22] [DABC.10. 9.22] [CDAB.15.14.23] [BCDA. 4.20.24]

[ABCD. 9.5.25] [DABC.14. 9.26] [CDAB. 3.14.27] [BCDA. 8.20.28]

[ABCD.13.5.29] [DABC. 2. 9.30] [CDAB. 7.14.21] [BCDA.12.20.32]

[ABCD. 5.4.33] [DABC. 8.11.34] [CDAB.11.16.35] [BCDA.14.23.36]

[ABCD. 1.4.37] [DABC. 4.11.38] [CDAB. 7.16.39] [BCDA.10.23.40]

[ABCD.13.4.41] [DABC. 0.11.42] [CDAB. 3.16.43] [BCDA. 6.23.44]

[ABCD. 9.4.45] [DABC.12.11.46] [CDAB.15.16.45] [BCDA. 2.23.48]

[ABCD. 0.6.49] [DABC. 7.10.50] [CDAB. 5.15.51] [BCDA. 5.21. 5]

[ABCD.12.6.53] [DABC. 3.10.54] [CDAB. 1.15.55] [BCDA. 1.21.56]

[ABCD. 8.6.57] [DABC.15.10.58] [CDAB.13.15.59] [BCDA.13.21.60]

[ABCD. 4.6.61] [DABC.11.10.62] [CDAB. 9.15.63] [BCDA. 9.21.64]

$A \leftarrow A+AA; B \leftarrow B+BB; C \leftarrow C+CC; D \leftarrow D+DD;$

**output:**  $(A, B, C, D)$

round 1      round 2      round 3      round 4

# Sicurezza di MD5

MD5 è stata oggetto di attacchi

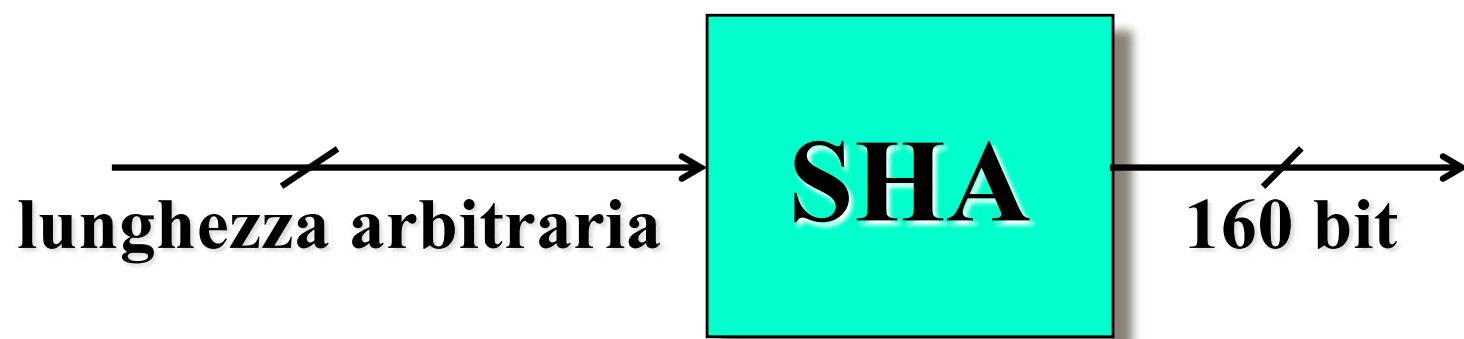
- 2004: Wang, Feng, Lai e Yu trovarono una collisione per MD5 (CRYPTO 2004)
- 2005: Lenstra, Wang, e de Weger mostrarono la costruzione di due certificati X.509 con differenti chiavi pubbliche e lo stesso MD5
- Vlastimil Klima, "Finding MD5 Collisions - a Toy For a Notebook", ePrint 2005/075
- Vlastimil Klima, "Tunnels in Hash Functions: MD5 Collisions Within a Minute", ePrint 2006/105

# Sicurezza di MD5

The screenshot shows the homepage of the CERT Vulnerability Notes Database. At the top, there are logos for CERT, Software Engineering Institute, and Carnegie Mellon. Below the logo, the title "Vulnerability Notes Database" is displayed in large blue text, followed by the subtitle "Advisory and mitigation information about software vulnerabilities". A navigation bar at the bottom has four items: "DATABASE HOME", "SEARCH", "REPORT A VULNERABILITY", and "HELP". The main content area features a red header "Vulnerability Note VU#836068" and the title "MD5 vulnerable to collision attacks". Below this, it says "Original Release date: 31 dic 2008 | Last revised: 21 gen 2009". A bold section titled "Do not use the MD5 algorithm" contains a warning message: "Software developers, Certification Authorities, website owners, and users should avoid using the MD5 algorithm in any capacity. As previous research has demonstrated, it should be considered cryptographically broken and unsuitable for further use."

# SHS

- SHS per **Secure Hash Standard**
- SHA per **Secure Hash Algorithm**
- Standard del Governo americano dal 1993
- Modificato nel luglio 1994, denotato **SHA-1**
  - (unica differenza: aggiunta di uno shift nell'espansione dei blocchi)
- Operazioni efficienti su architetture 32 bit big-endian
- Stessi principi di MD5, ma più sicuro



# SHA: padding del messaggio

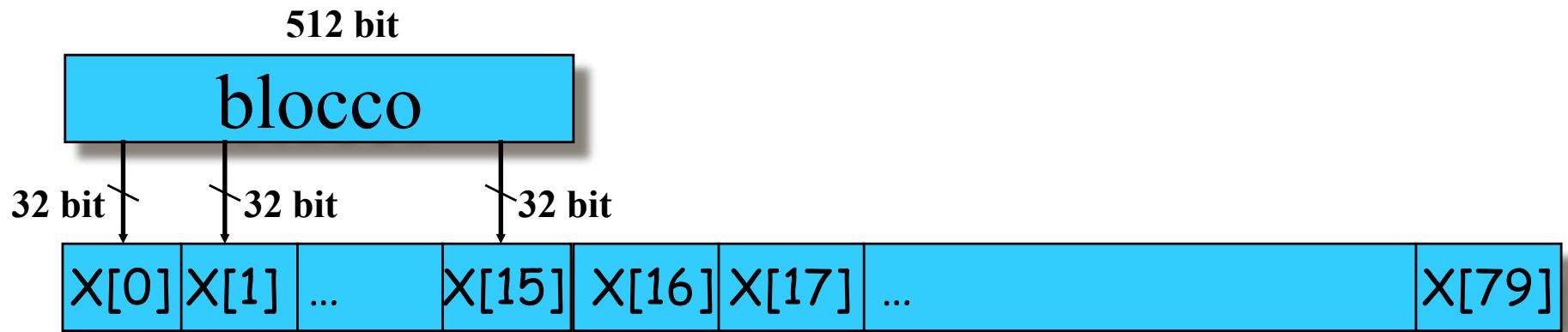
- SHA processa il messaggio in blocchi di 512 bit
  - Ogni blocco consta di 16 parole di 32 bit
- M messaggio originario di b bit → padding

$$M' = \boxed{M \ 100\ldots0 \ b}$$

(447-b) mod 512 bit      64 bit

- $M'$  consta di un numero di bit multiplo di 512
  - ovvero di un numero di parole  $N$  multiplo di 16
  - quindi,  $N/16$  blocchi di 512 bit

# Espansione blocco ed Iterazioni



$$X[t] \leftarrow (X[t-3] \oplus X[t-8] \oplus X[t-14] \oplus X[t-16]) \ll 1$$

- 80 iterazioni
- Una parola ed una costante per ogni iterazione

# Funzioni logiche di SHA

Funzione  $F(t, X, Y, Z)$

round  $t = 0, \dots, 19$ :  $F(t, X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$  (if  $X$  then  $Y$  else  $Z$ )

round  $t = 20, \dots, 39$ :  $F(t, X, Y, Z) = X \oplus Y \oplus Z$  (bit di parità)

round  $t = 40, \dots, 59$ :  $F(t, X, Y, Z) = (X \wedge Z) \vee (Y \wedge Z) \vee (X \wedge Y)$  (2 su 3)

round  $t = 60, \dots, 79$ :  $F(t, X, Y, Z) = Y \oplus X \oplus Z$  (bit di parità)

X	Y	Z	F(0,...)	F(20,...)	F(40,...)	F(60,...)
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	0	1	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	1	0	0	1	0
1	1	0	1	0	1	0
1	1	1	1	1	1	1

# Costanti additive di SHA

- Costante additiva  $K[t]$ :
  - round  $t = 0, \dots, 19$ : 5a827999
  - round  $t = 20, \dots, 39$ : 6ed9eba1
  - round  $t = 40, \dots, 59$ : 8f1bbcdcc
  - round  $t = 60, \dots, 79$ : ca62c1d1

```

A=67452310; B=efcdab89; C=98badcfe; D=10325476; E=c3d2e1f0;
for i = 0 to N/16-1 do
    for j = 0 to 15 do
        X[j] ← M'[16i+j]
    for t = 16 to 79 do
        X[t] ← ( X[t-3] ⊕ X[t-8] ⊕ X[t-14] ⊕ X[t-16] ) ≪ 1
        AA ← A; BB ← B; CC ← C; DD ← D; EE ← E;
        for t=0 to 79 do
            TEMP ← (A<<5) + F(t,B,C,D) + E + X[t] + K[t]
            E ← D
            D ← C
            C ← (B<<30)
            B ← A
            A ← TEMP
            A ← A + AA; B ← B + BB; C ← C + CC; D ← D + DD; E ← E + EE;
output: (A, B, C, D, E)

```

**SHA-1**

espansione  
da 16 ad 80 parole  
“<<1” non c’era in SHA

# Sicurezza di SHA-1

## Miglior attacco

- Complessità stimata  $2^{57,5}$
- Marc Stevens, *New collision attacks on SHA-1 based on optimal joint local-collision analysis*, Eurocrypt 2013

# Raccomandazione Microsoft

Security TechCenter > Security Advisories > Microsoft Security Advisory (2880823)

## Microsoft Security Advisory (2880823)

### Deprecation of SHA-1 Hashing Algorithm for Microsoft Root Certificate Program

Published: Tuesday, November 12, 2013

Version: 1.0

#### General Information

##### Executive Summary

Microsoft is announcing a policy change to the Microsoft Root Certificate Program. The new policy will no longer allow root certificate authorities to issue X.509 certificates using the SHA-1 hashing algorithm for the purposes of SSL and code signing after January 1, 2016. Using the SHA-1 hashing algorithm in digital certificates could allow an attacker to spoof content, perform phishing attacks, or perform man-in-the-middle attacks.

**Recommendation:** Microsoft recommends that certificate authorities no longer sign newly generated certificates using the SHA-1 hashing algorithm and begin migrating to SHA-2. Microsoft also recommends that customers replace their SHA-1 certificates with SHA-2 certificates at the earliest opportunity. Please see the **Suggested Actions** section of this advisory for more information.

# Sicurezza di SHA-1

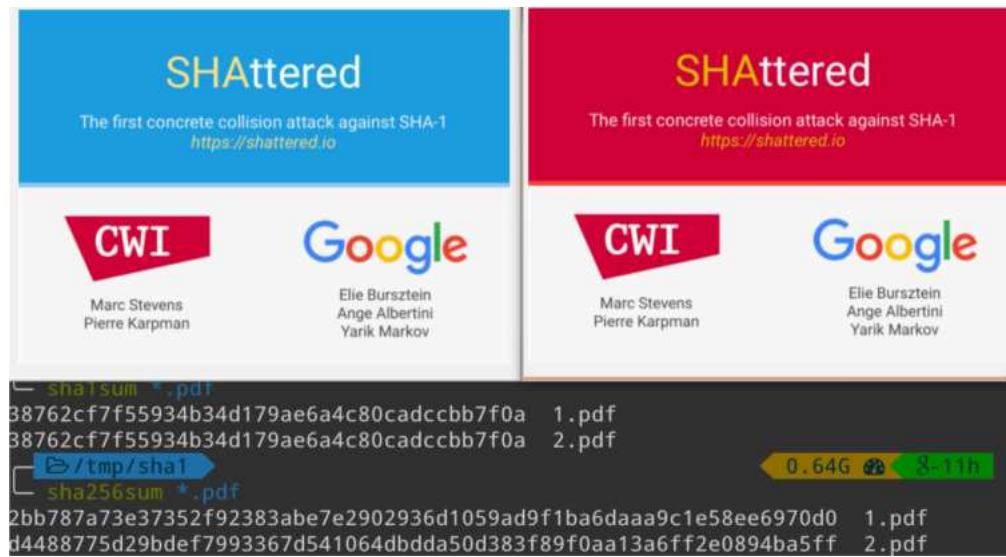
## Miglior attacco

- Complessità stimata  $2^{57,5}$
- Marc Stevens, *New collision attacks on SHA-1 based on optimal joint local-collision analysis*, Eurocrypt 2013
- Nota: le collisioni sono semplici stringhe binarie non file in un certo formato

# SHAttered attack

Trovati 2 differenti file pdf con lo stesso valore hash SHA-1 con circa  $2^{63.1}$  valutazioni di SHA-1

- 9,223,372,036,854,775,808 SHA1 computations in total
- 6,500 years of CPU computation to complete the attack first phase
- 110 years of GPU computation to complete the second phase
- CWI (Centrum Wiskunde & Informatica) e Google
- 23 Febbraio 2017



# Federal Information Processing Standard SHA

- SHA
  - FIPS 180 Secure Hash Standard (maggio 1993)
- SHA-1
  - FIPS 180-1 Secure Hash Standard (aprile 1995)
- SHA-2
  - FIPS 180-2 Secure Hash Standard (agosto 2002)  
aggiunge a SHA-1: **SHA-224, SHA-256, SHA-384, SHA-512**
  - FIPS 180-3 Secure Hash Standard (ottobre 2008)
  - FIPS 180-4 Secure Hash Standard (marzo 2012)  
aggiunge **SHA-512/224 e SHA-512/256**
- SHA-3
  - FIPS 202 (Agosto 2015)
  - **SHA3-224, SHA3-256, SHA3-384, e SHA3-512**
  - Aggiunge 2 extendable-output functions (XOFs): **SHAKE128 e SHAKE256**

# SHA-2

- Stessi principi di MD4, MD5, SHA-1

Algorithm	Message Size (bits)	Block Size (bits)	Word Size (bits)	Message Digest Size (bits)
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-512	$< 2^{128}$	1024	64	512
SHA-512/224	$< 2^{128}$	1024	64	224
SHA-512/256	$< 2^{128}$	1024	64	256

- SHA-384

- Valore hash = primi 384 bit di SHA-512, con costanti iniziali cambiate

# SHA-2

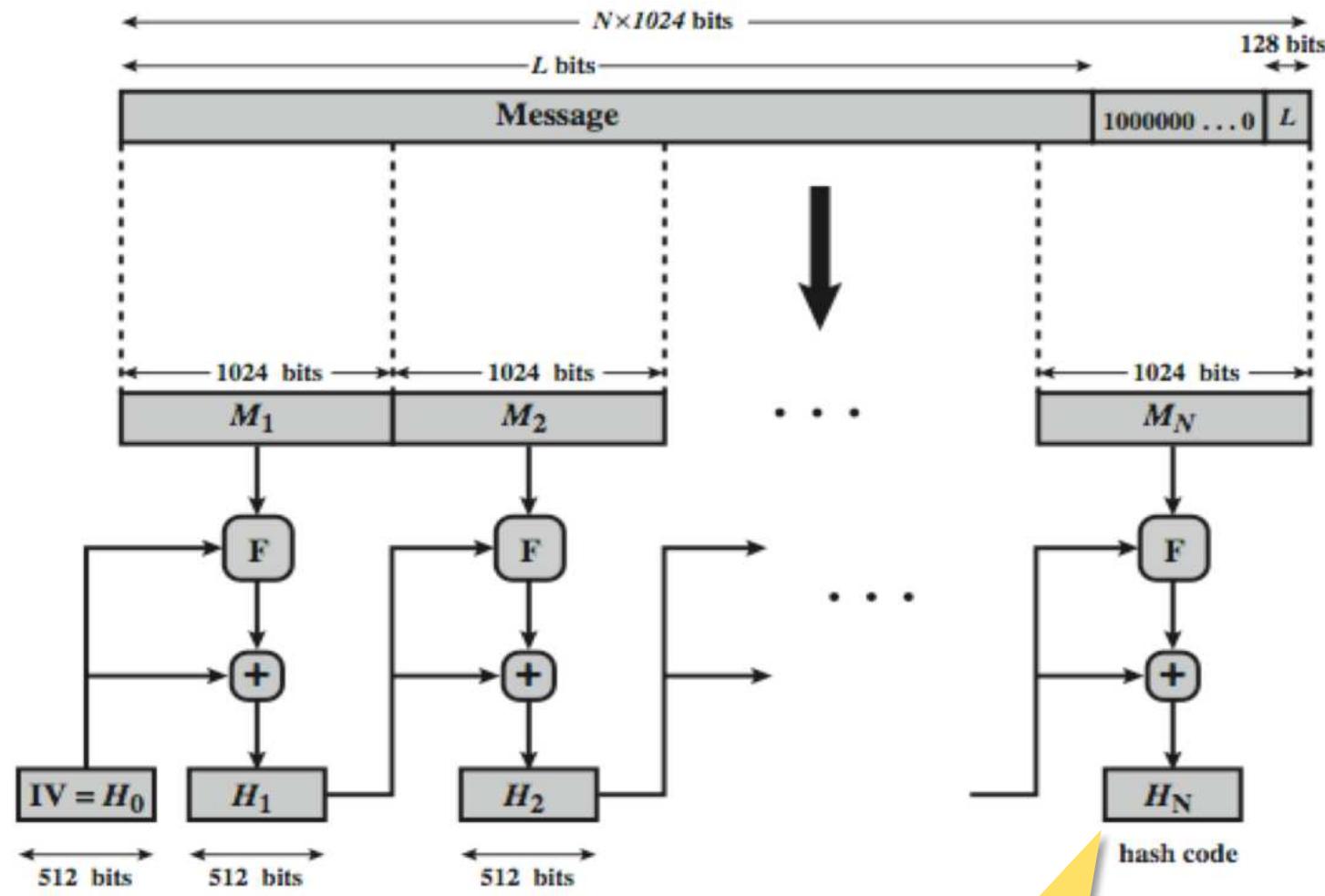
	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
<b>Collision Resistance Strength in bits</b>	< 80 <sup>3</sup>	112	128	192	256
<b>Preimage Resistance Strength in bits</b>	160	224	256	384	512
<b>Second Preimage Resistance Strength in bits</b>	105-160	201-224	201-256	384	394-512

Table 1: Strengths of the Security Properties of the Approved Hash Algorithms

<sup>3</sup> Current estimated value is around 60.

NIST Special Publication 800-107, Recommendation for Applications Using Approved Hash Algorithms, Feb 2009

# SHA-512 Overview



$+$  = word-by-word addition mod  $2^{64}$

valore hash

# SHA-3

- Annuncio di attacco [Wang 2005] che trova collisioni in  $2^{63}$  e non  $2^{80}$  per SHA-1
- Nonostante nessuno lo abbia verificato, inizia la scelta di una nuova funzione hash, chiamata SHA-3
- Competizione pubblica simile a quella per AES
- Requisiti minimi, richieste per la sottomissione, criteri valutazione, versione draft sul Federal Register Notice (23 gennaio 2007)
- Commenti pubblici entro il 27 aprile 2007
- Call for a New Cryptographic Hash Algorithm (SHA-3) Family sul Federal Register Notice (2 novembre 2007)
- Sottomissione proposte entro 31 ottobre 2008
- Ricevute 64 proposte

# SHA-3 (Round 1)

- Annunciati 51 candidati per il Round 1, che rispettavano i requisiti minimi (9 dicembre 2008)

Abacus • ARIRANG • AURORA • BLAKE • Blender • Blue Midnight Wish • BOOLE • Cheetah • CHI CRUNCH • CubeHash • DCH • Dynamic SHA • Dynamic SHA2 • ECHO • ECOH • EDON-R • EnRUPT ESSENCE • FSB • Fugue • Grøstl • Hamsi • JH • Keccak • Khichidi-1 • LANE • Lesamnta • Luffa • LUX • MCSSHA-3 • MD6 • MeshHash • NaSHA • SANDstorm • Sarmal • Sgàil • Shabal • SHAMATA SHAvite-3 SIMD Skein • Spectral Hash • StreamHash • SWIFFTX • Tangle • TIB3 • Twister • Vortex • WaMM • Waterfall

- First SHA-3 Candidate Conference, 25-28 febbraio 2009, Katholieke Universiteit Leuven, Belgio

"the purpose is to allow the submitters of the first round candidates to present their algorithms, and for NIST to discuss the way forward with the competition."

# SHA-3 (Round 2)

- Annunciati 14 candidati per il Round 2  
(24 luglio 2009)

BLAKE • Blue Midnight Wish • CubeHash • ECHO • Fugue • Grøstl •  
Hamsi • JH • Keccak • Luffa • Shabal • SHAvite-3 • SIMD • Skein

- Second SHA-3 Candidate Conference,  
23-24 agosto 2010, Santa Barbara

# SHA-3 (Final Round, timeline)

- Annunciati 5 candidati per il Round 3, (9 dic 2010)

BLAKE • Grøstl • JH • Keccak • Skein

- Un anno di commenti pubblici
- Final SHA-3 Candidate Conference, (Washington, 22-23 marzo 2012)
- Proclamazione vincitore (2 ott 2012)      Keccak
- DRAFT FIPS 202 SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions (4 aprile 2014)
- Commenti pubblici (90 giorni)
- SHA-3 Workshop, Santa Barbara, 22-23 agosto 2014
- FIPS 202 pubblicato (agosto 2015)

# FIPS 202

- SHA3-224, SHA3-256, SHA3-384, SHA3-512
- Extendable-output functions **SHAKE128** **SHAKE256**

Function	Output Size	Security Strengths in Bits		
		Collision	Preimage	2nd Preimage
SHA-1	160	< 80	160	$160 - L(M)$
SHA-224	224	112	224	$\min(224, 256 - L(M))$
SHA-512/224	224	112	224	224
SHA-256	256	128	256	$256 - L(M)$
SHA-512/256	256	128	256	256
SHA-384	384	192	384	384
SHA-512	512	256	512	$512 - L(M)$
SHA3-224	224	112	224	224
SHA3-256	256	128	256	256
SHA3-384	384	192	384	384
SHA3-512	512	256	512	512
SHAKE128	$d$	$\min(d/2, 128)$	$\geq \min(d, 128)$	$\min(d, 128)$
SHAKE256	$d$	$\min(d/2, 256)$	$\geq \min(d, 256)$	$\min(d, 256)$

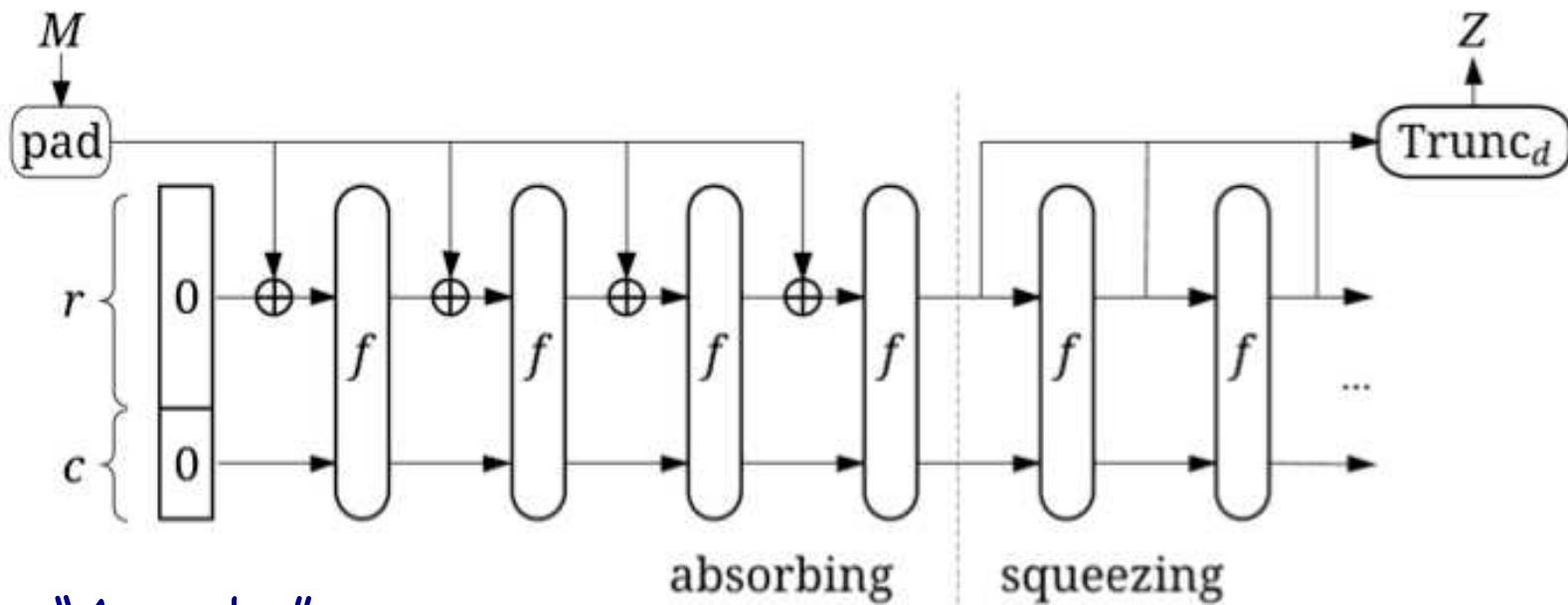
$$L(M) = \left\lceil \log_2 \left( \frac{\text{len}(M)}{B} \right) \right\rceil$$

B è la lunghezza blocco:  
 512 per SHA-1, SHA-224, ...  
 1024 per SHA512

Table 3: Security strengths of SHA-1, SHA-2, and SHA-3 functions

# Sponge Construction

SPONGE[ $f$ ,pad,r]( $M,d$ )



"Assorbe" un numero

arbitrario di bit nel suo stato

e poi "spreme" un numero

arbitrario di bit dal suo  
stato



# Sponge Construction

## KECCAK

$\text{KECCAK}[c](M, d)$   
= SPONGE[ $\text{KECCAK-p[1600,24]}$ , $\text{pad10*1,1600-c}$ ]( $M, d$ )

# Sponge Construction

## KECCAK

KECCAK[c](M, d)

= SPONGE[KECCAK-p[1600,24], pad10\*1, 1600-c](M, d)

SHA3-224(M) = KECCAK[448](M || 01, 224)

SHA3-256(M) = KECCAK[512](M || 01, 256)

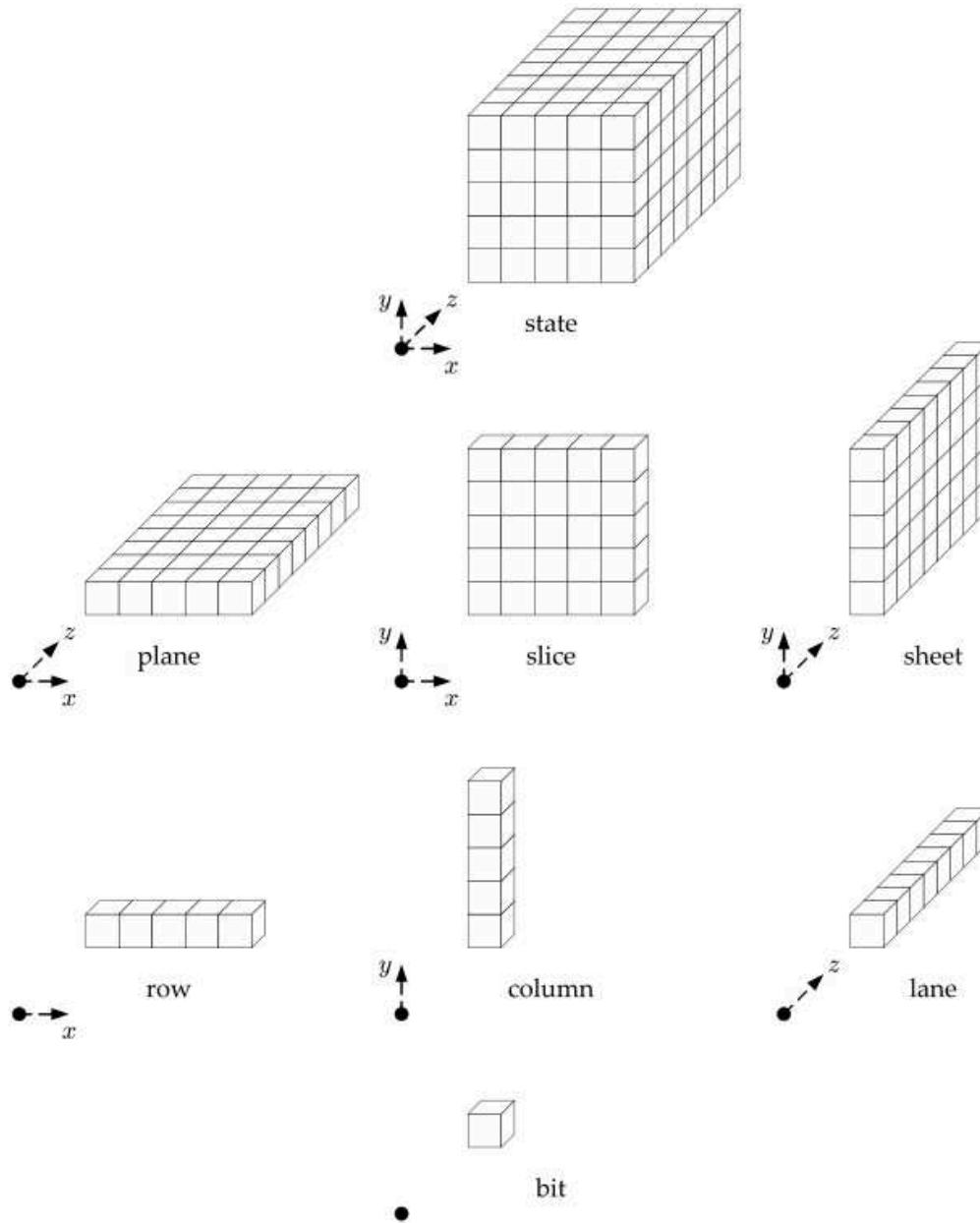
SHA3-384(M) = KECCAK[768](M || 01, 384)

SHA3-512(M) = KECCAK[1024](M || 01, 512)

SHAKE128(M, d) = KECCAK[256](M || 1111, d)

SHAKE256(M, d) = KECCAK[512](M || 1111, d)

# KECCAK: stato e parti



# KECCAK-p: funzioni $\theta, \rho, \pi, \chi, \iota$

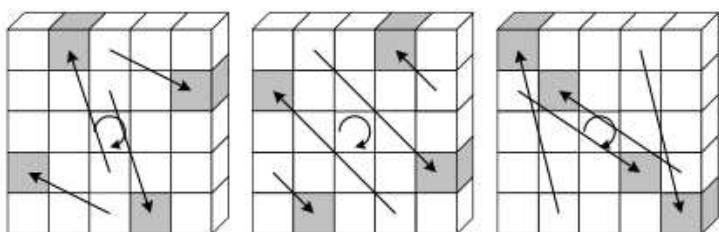
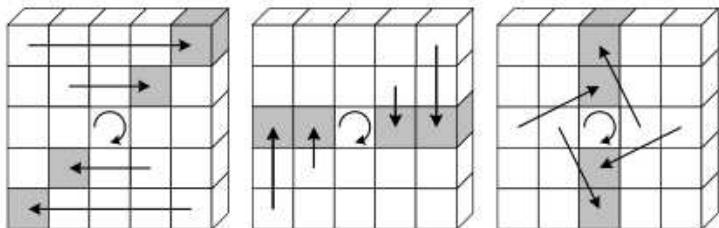


Illustration of  $\pi$  applied to a single slice [8]

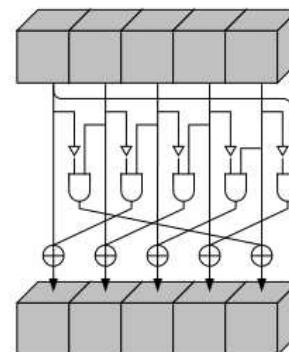


Illustration of  $\chi$  applied to a single row

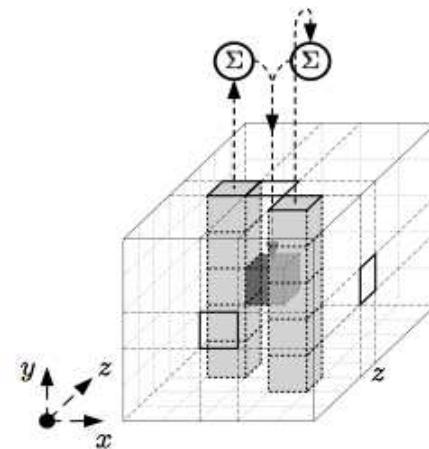
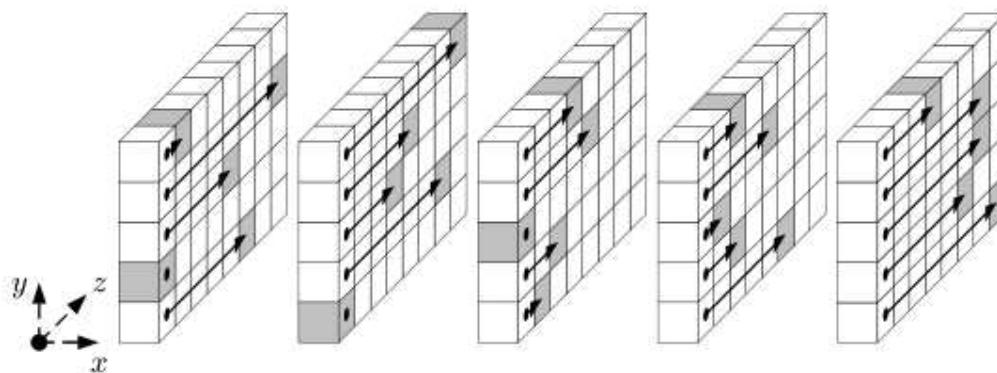


Illustration of  $\theta$  applied to a single bit



usa linear feedback shift register

$$(x^t \bmod x^8 + x^6 + x^5 + x^4 + 1) \bmod x \text{ in GF}(2)$$

# NIST Policy on Hash Functions

August 5, 2015

**SHA-1:** Federal agencies **should** stop using SHA-1 for generating digital signatures, generating time stamps and for other applications that require collision resistance. Federal agencies may use SHA-1 for the following applications: verifying old digital signatures and time stamps, generating and verifying hash-based message authentication codes (HMACs), key derivation functions (KDFs), and random bit/number generation. Further guidance on the use of SHA-1 is provided in SP 800-131A.

**SHA-2 (i.e., SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256):** Federal agencies may use these hash functions for all applications that employ secure hash algorithms. NIST encourages application and protocol designers to implement SHA-256 at a minimum for any applications of hash functions requiring interoperability. Further guidance on the use of SHA-2 is provided in SP 800-57 Part 1, section 5.6.2 and SP 800-131A.

**SHA-3 (i.e., SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128 and SHAKE256):** Federal agencies may use the four fixed-length SHA-3 algorithms—**SHA3-224, SHA3-256, SHA3-384, and SHA3-512** for all applications that employ secure hash algorithms. The SHA-3 Extendable-Output Functions (XOFs), **SHAKE128** and **SHAKE256**, can be specialized to hash functions, subject to additional security considerations. Guideline for using the XOFs will be provided in the future. Currently there is no need to transition applications from SHA-2 to SHA-3.

# SHA-3: Bibliografia

## SHA-3 COMPETITION

<http://csrc.nist.gov/groups/ST/hash/sha-3/index.html>

The screenshot shows the NIST Computer Security Division (CSD) website. The header includes the NIST logo, the text "National Institute of Standards and Technology Information Technology Laboratory", a search bar, and links for "ABOUT", "MISSION", "CONTACT", "STAFF", and "SITE MAP". The main navigation menu below the header includes "CSRC HOME", "GROUPS", "PUBLICATIONS", "DRIVERS", "FEDERAL REGISTER NOTICES", "NEWS & EVENTS", and "ARCHIVE". The page title is "Computer Security Division CSD Computer Security Resource Center". The left sidebar contains links for "Cryptographic Hash & SHA-3 Standard Development", "Pre-SHA3 Competition (2004-2007)", "SHA-3 Competition (2007-2012)" (which is highlighted in blue), "Submission Requirements", "Round 1", "Round 2", "Round 3", "SHA-3 Standardization (2013- )", "NIST Policy on Hash Functions", "Hash Forum", and "Contacts". The main content area discusses the "SHA-3 COMPETITION (2007-2012)". It states that NIST announced a competition in a Federal Register Notice on November 2, 2007, to develop a new cryptographic hash algorithm called SHA-3. It mentions the submission of sixty-four entries, selection of fifty-one first-round candidates, fourteen second-round candidates, and five finalists (BLAKE, Grøstl, JH, Keccak, and Skein). It also notes the advancement of these candidates to the third and final round. The text concludes by mentioning the public feedback and the announcement of Keccak as the winner on October 2, 2012. At the bottom, there are links for "Hash Project Webmaster", "Disclaimer Notice & Privacy Policy", and "NIST is an Agency of the U.S. Department of Commerce". The footer also includes update and creation dates: "Last updated: March 31, 2014" and "Page created: April 15, 2005".

# Calcolo hash online: un esempio

## SHA256 Hash Generator

This online tool allows you to generate the SHA256 hash of any string. SHA256 is designed by NSA, it's more reliable than SHA1.

Enter your text below:

Generate   Clear All   MD5   SHA1   SHA512   Password Generator

Treat each line as a separate string

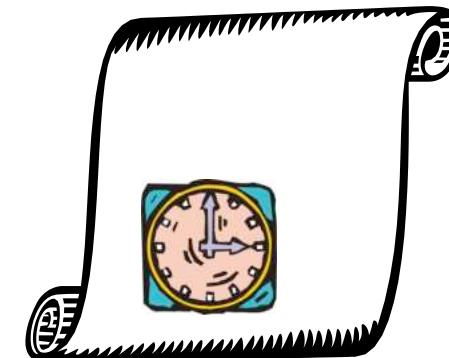
SHA256 Hash of your string:

3340AE93D3ACADC4BFAD32C67F9498E8350FA94DAB19CD70C15D882FCF06546A|

<https://passwordsgenerator.net/sha256-hash-generator/>

# Marcatura temporale di documenti digitali

- Il notaio digitale
- Quando è stato creato



il documento D ?

# Digital Timestamp

La *marca temporale* di un documento è qualcosa aggiunto ad esso che prova che il documento è stato "prodotto" prima, dopo oppure ad un fissato momento

# Alcune idee

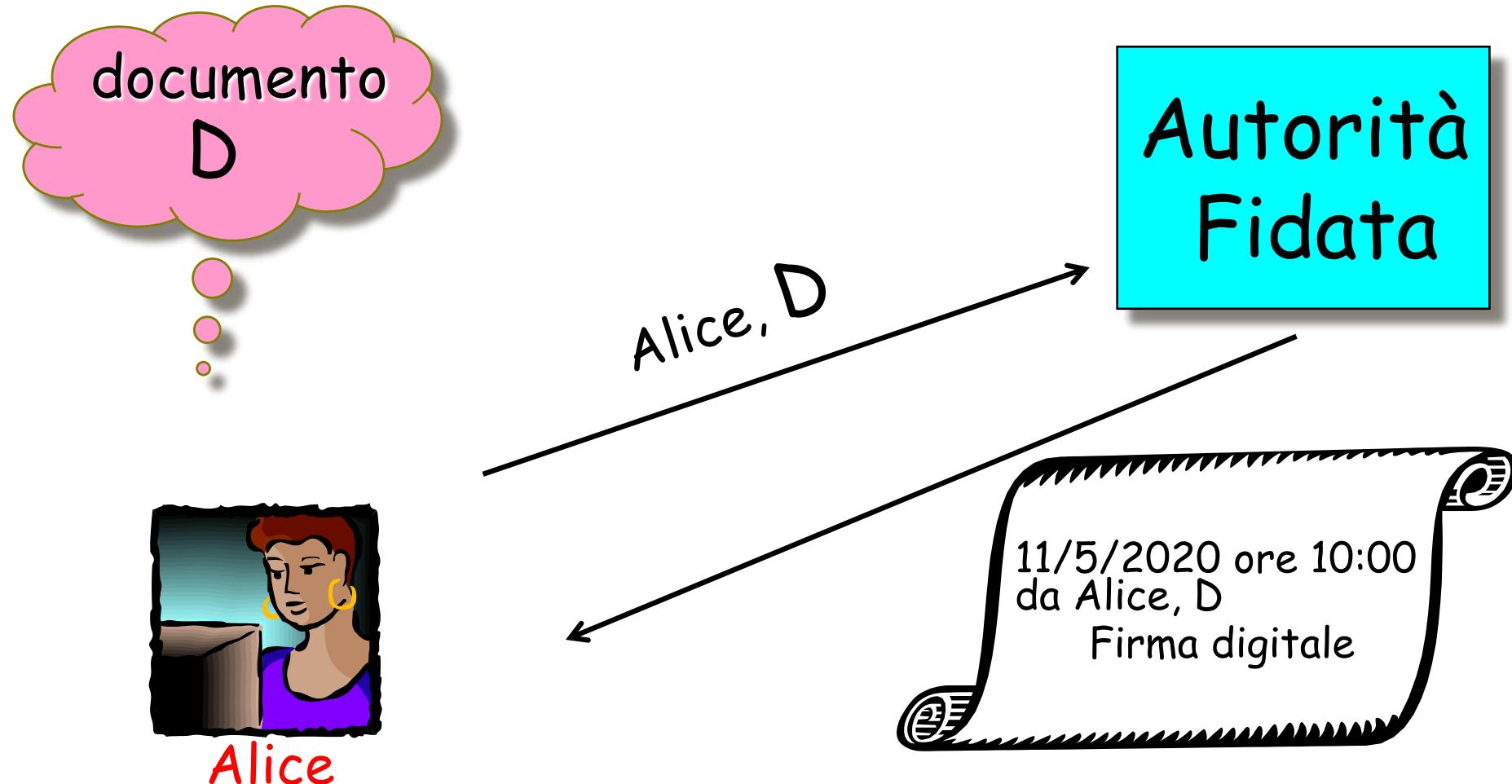
- Depositare il documento presso un notaio
- Inviare il documento a se stesso, tramite il servizio postale
- Brevetto (se brevettabile... )
- Pubblicare il documento su di un giornale
- Uso di un registro di protocollo
- Foto con un quotidiano (se è un sequestro... )



# Facile e Difficile

- È in genere *facile* provare che un documento è stato prodotto *dopo* una data fissata
- È in genere *difficile* provare che un documento è stato prodotto prima di una data fissata

# Una prima soluzione



# Problemi con la soluzione *naive*

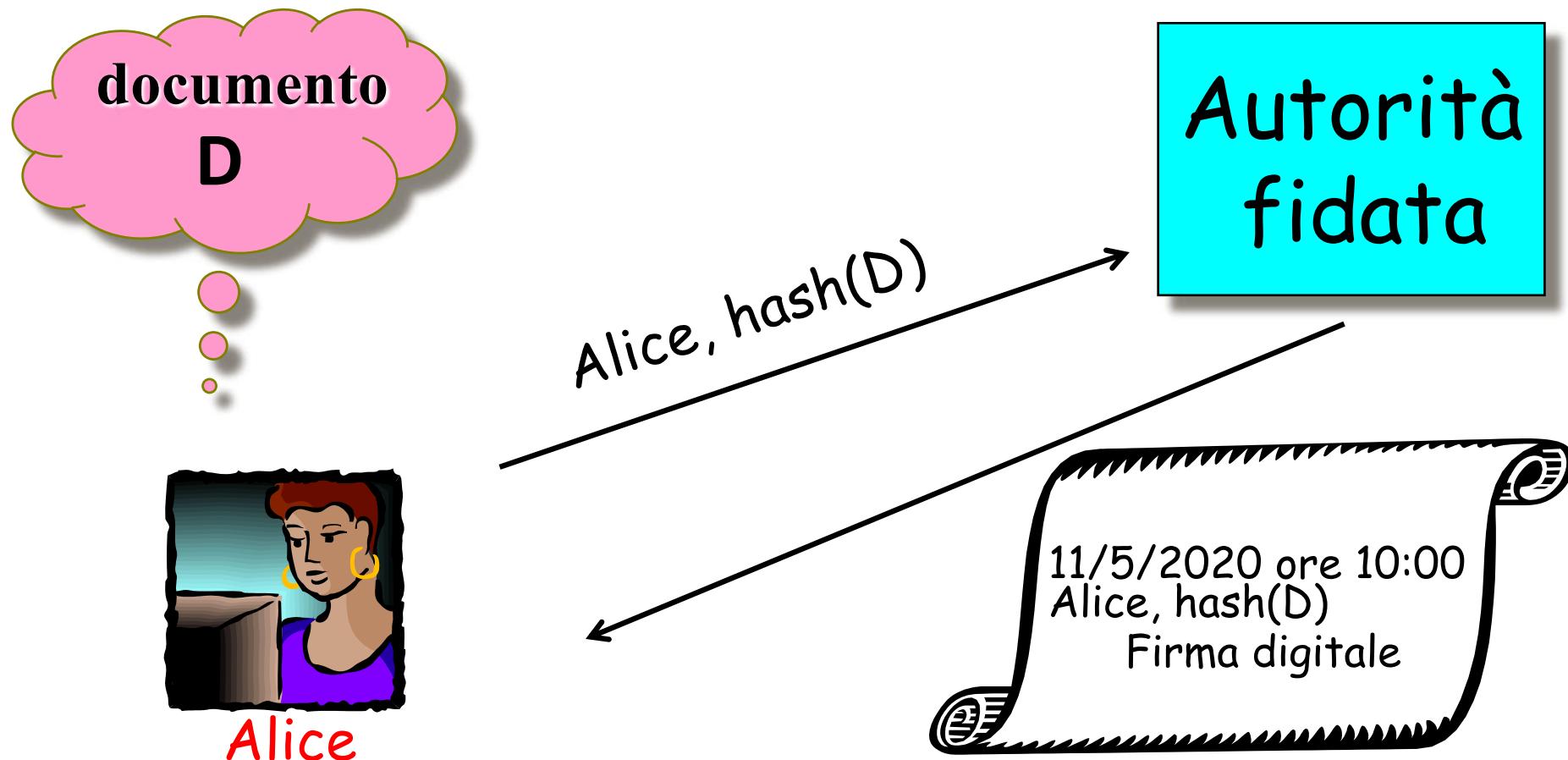
- Dimensioni del documento D
  - per la comunicazione
  - per la memorizzazione dell'Autorità Fidata
- Privatezza del contenuto di D
- Quanto è fidata l'Autorità Fidata?

# Idea: Funzioni Hash



- Dimensioni del documento D
  - per la comunicazione
  - per la memorizzazione dell'Autorità Fidata
- Privatezza del contenuto di D
- Quanto è fidata l'Autorità Fidata?

# Soluzione migliorata



# Marca temporale in Italia

- Quadro normativo
- Caratteristiche
- Utilità

# Marca temporale

- “evidenza informatica che consente di rendere opponibile a terzi un riferimento temporale” art. 1 del D.M.E.F. 23 gennaio 2004
- “il riferimento temporale che consente la validazione temporale” art. 1 lett. i) del D.P.C.M. 30 marzo 2009
- “validazione temporale” è il “risultato della procedura informatica con cui si attribuiscono, ad uno o più documenti informatici, una data ed un orario opponibili ai terzi” art. 1, lett. aa), del CAD

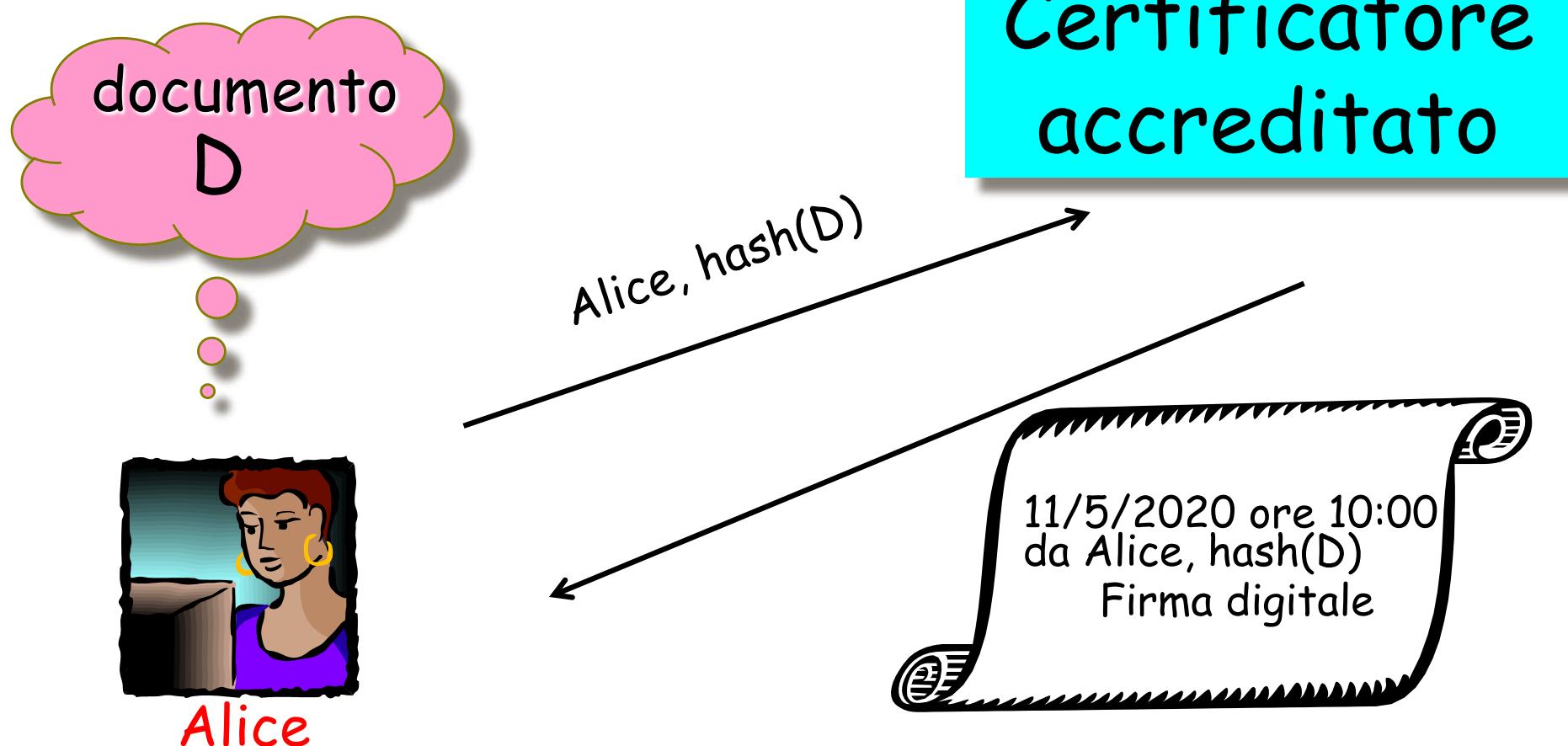
# Codice dell'Amministrazione Digitale (CAD)

Decreto legislativo 7 marzo 2005 n. 82

Modifiche successive:

- Decreto legislativo 4 aprile 2006, n. 159
- Decreto legge n. 185/2008, convertito in Legge n. 2/2009
- Legge 18 giugno 2009, n. 69
- Legge 3 agosto 2009, n. 102
- Decreto legislativo 30 dicembre 2010, n. 235
- Decreto legge 21 giugno 2013 n. 69, convertito con modificazioni dalla L. 9 agosto 2013, n. 98

# Marca temporale: realizzazione



# Marca temporale: tempo

Art. 47. (Precisione dei sistemi di validazione temporale) del D.P.C.M. 30 marzo 2009

1. Il riferimento temporale assegnato ad una marca temporale coincide con il momento della sua generazione, con una differenza non superiore ad un minuto secondo rispetto alla scala di tempo UTC (IEN), di cui al decreto del Ministro dell'industria, del commercio e dell'artigianato 30 novembre 1993, n. 591
2. Il riferimento temporale contenuto nella marca temporale è specificato con riferimento al Tempo Universale Coordinato (UTC)

# Marca temporale: tempo

Si basano su orologi atomici

- Tempo UTC
  - Tempo GPS
- Il tempo coordinato universale UTC è il fuso orario "0" da cui sono poi calcolati tutti gli altri fusi orari del mondo

UTC	2020-05-10 17:57:07
GPS	2020-05-10 17:57:25
LORAN	2020-05-10 17:57:34
TAI	2020-05-10 17:57:44

<http://www.ipses.com/prod/timing/UTC-GPS.php?language=it>

# Marca temporale: tempo

Si basano su orologi atomici

- Tempo UTC
- Tempo GPS

UTC	2020-05-10 17:57:07
GPS	2020-05-10 17:57:25
LORAN	2020-05-10 17:57:34
TAI	2020-05-10 17:57:44

<http://www.ipses.com/prod/timing/UTC-GPS.php?language=it>



[L'Istituto](#) [Ricerca](#) [Formazione](#) [Eventi](#) [Press Area](#) [Webmail](#) [Servizi interni](#) [Mobile](#)

**Contatti**  
Indirizzi - Rubrica

INRIM  
Strada delle Cacce 91  
10135 Torino, ITALY

C.F. / P.IVA : 09261710017

Posta Elettronica Certificata

[inrim@pec.it](mailto:inrim@pec.it)

L'Istituto Nazionale di Ricerca Metrologica (INRIM) è un ente pubblico di ricerca, afferente al Ministero dell'Istruzione, dell'Università e della Ricerca. Si occupa di scienza delle misure e dei materiali, sviluppa tecnologie e dispositivi innovativi. Adempiendo ai suoi compiti di istituto metrologico primario, l'INRIM realizza i campioni primari delle unità di misura fondamentali e derivate del Sistema Internazionale delle unità di misura (SI), ne assicura il mantenimento, partecipa ai confronti internazionali e permette in Italia la riferibilità di ogni misura al SI; rappresenta l'Italia negli organismi metrologici internazionali.

[leggi »](#)

**Tempo campione**

Ora esatta



Ora legale

Server NTP

# **Marca temporale: prolungamento validità documento nel tempo**

Art. 49. (Registrazione delle marche generate) del D.P.C.M. 30 marzo 2009

1. Tutte le marche temporali emesse da un sistema di validazione sono conservate in un apposito archivio digitale non modificabile per un periodo non inferiore a venti anni ovvero, su richiesta dell'interessato, per un periodo maggiore, alle condizioni previste dal certificatore.

# Uso Funzioni Hash

- Dimensioni del documento D
  - per la comunicazione
  - per la memorizzazione dell'Autorità Fidata
- Privatezza del contenuto di D
- Quanto è fidata l'Autorità Fidata?

# Problema

Sed quis custodiet  
ipsos custodes?

Giovenale, *Satire*, VI, 100 A.C.



# Possibili Soluzioni

Due famiglie di protocolli

**Protocolli distribuiti** (senza Autorità Fidata)



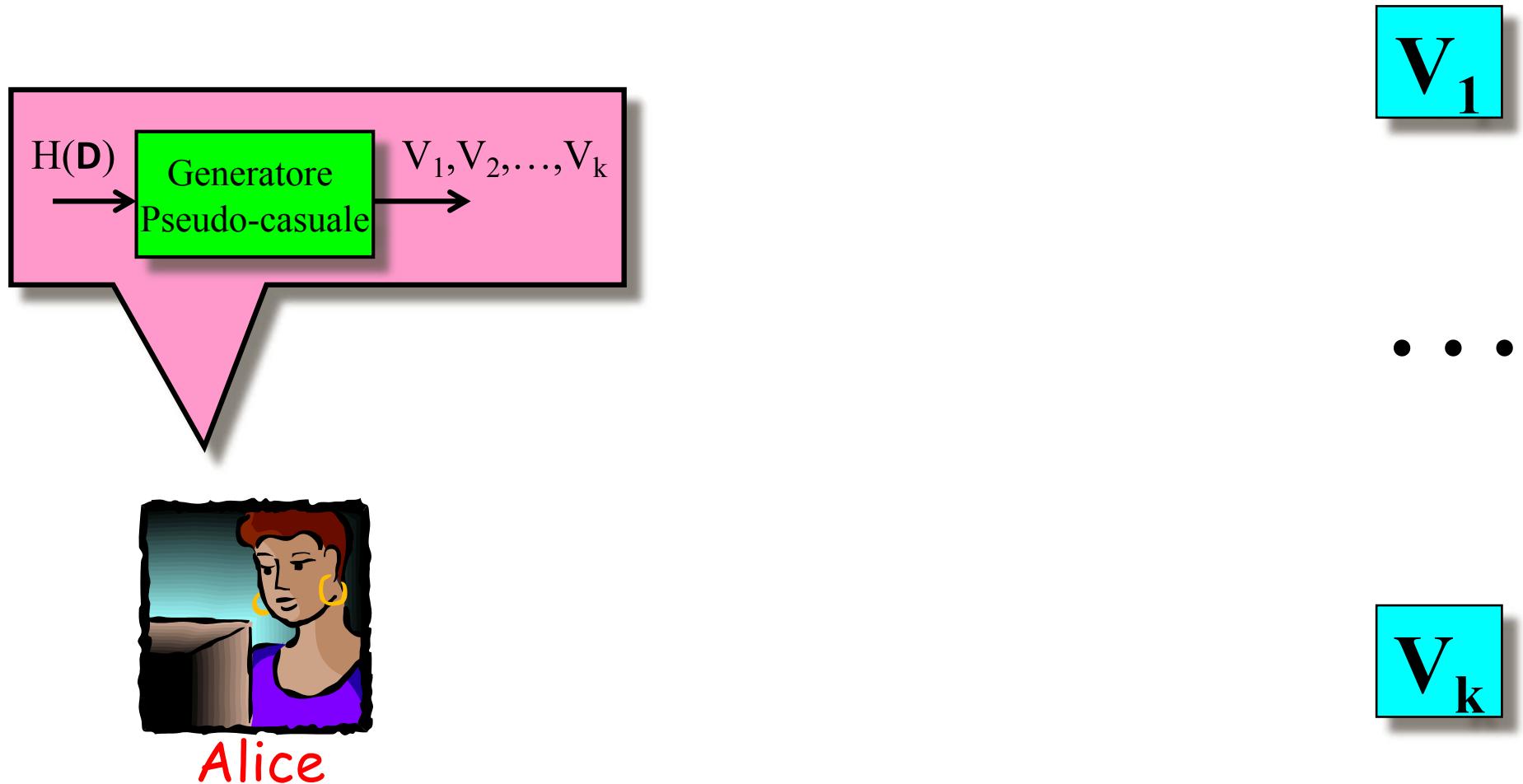
Avere più "testimonianze" del tempo

**Protocolli con "link"** (con Autorità Fidata)

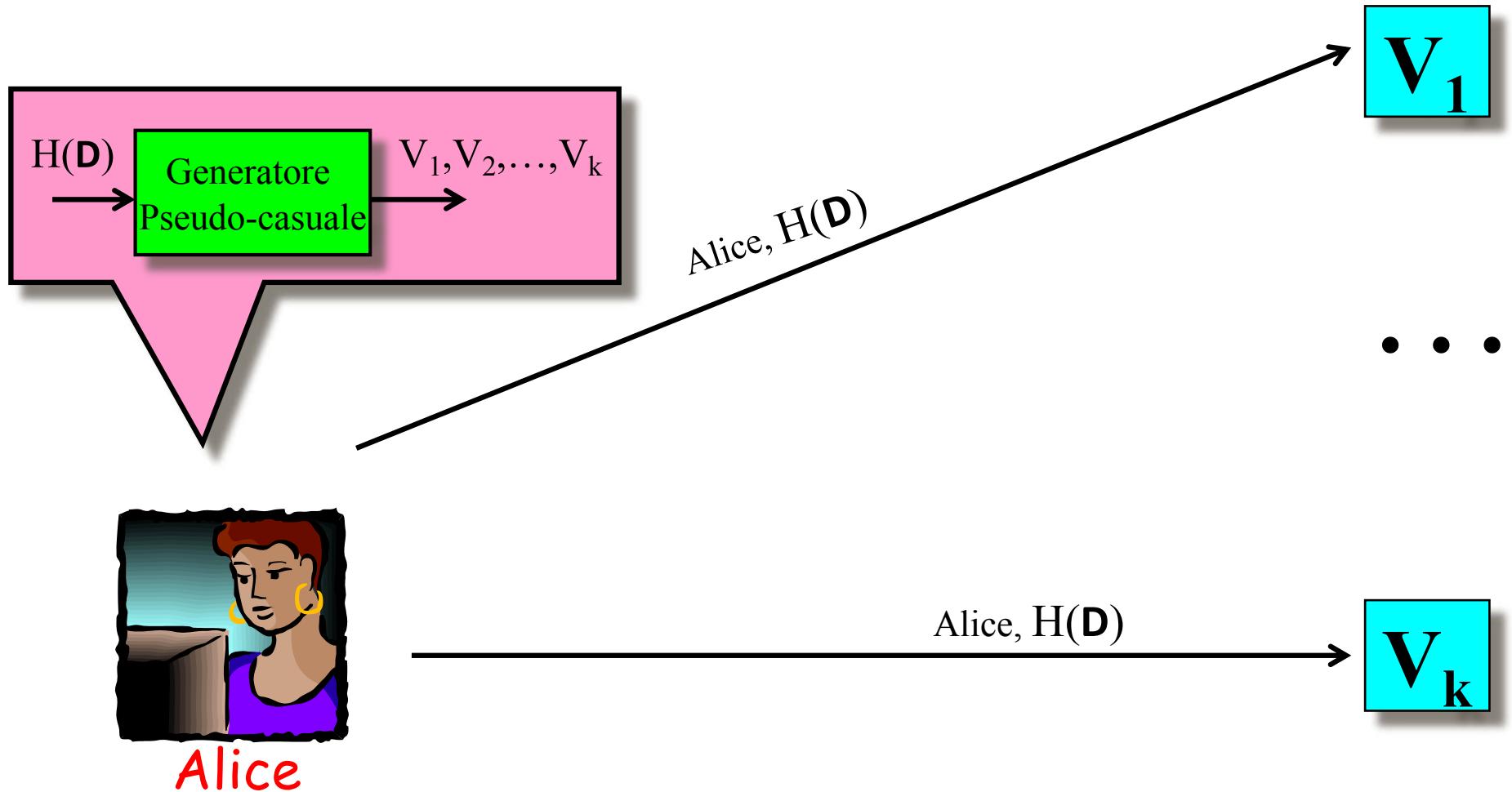


Collegare tra loro le marche  
dei documenti

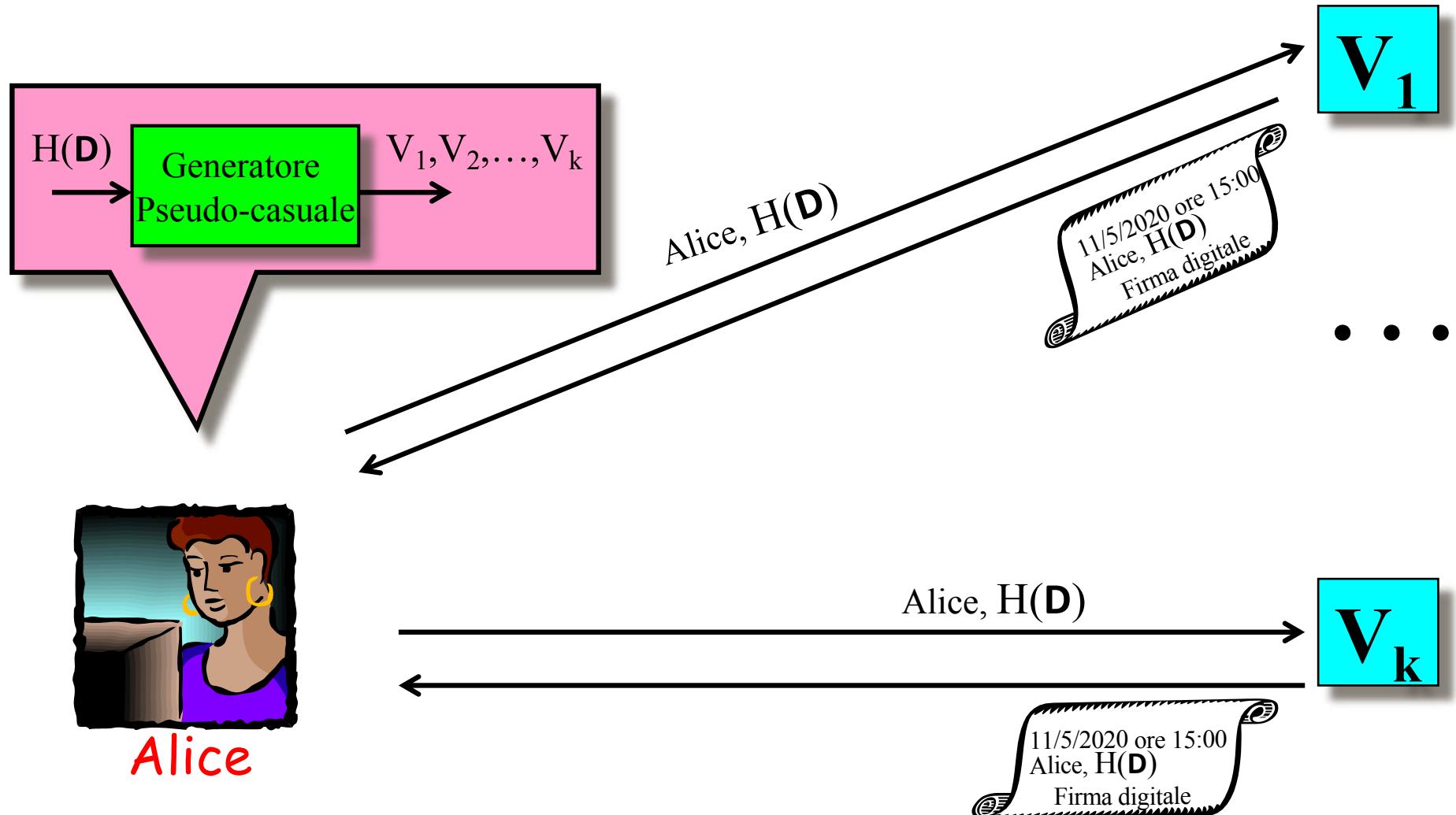
# Un protocollo distribuito



# Un protocollo distribuito



# Un protocollo distribuito



# Protocollo Distribuito: Sicurezza

$k$  grande  $\Rightarrow$  difficile per Alice  
corrompere  $k$  persone

La scelta delle persone da contattare è

- casuale
- dipendente dal documento

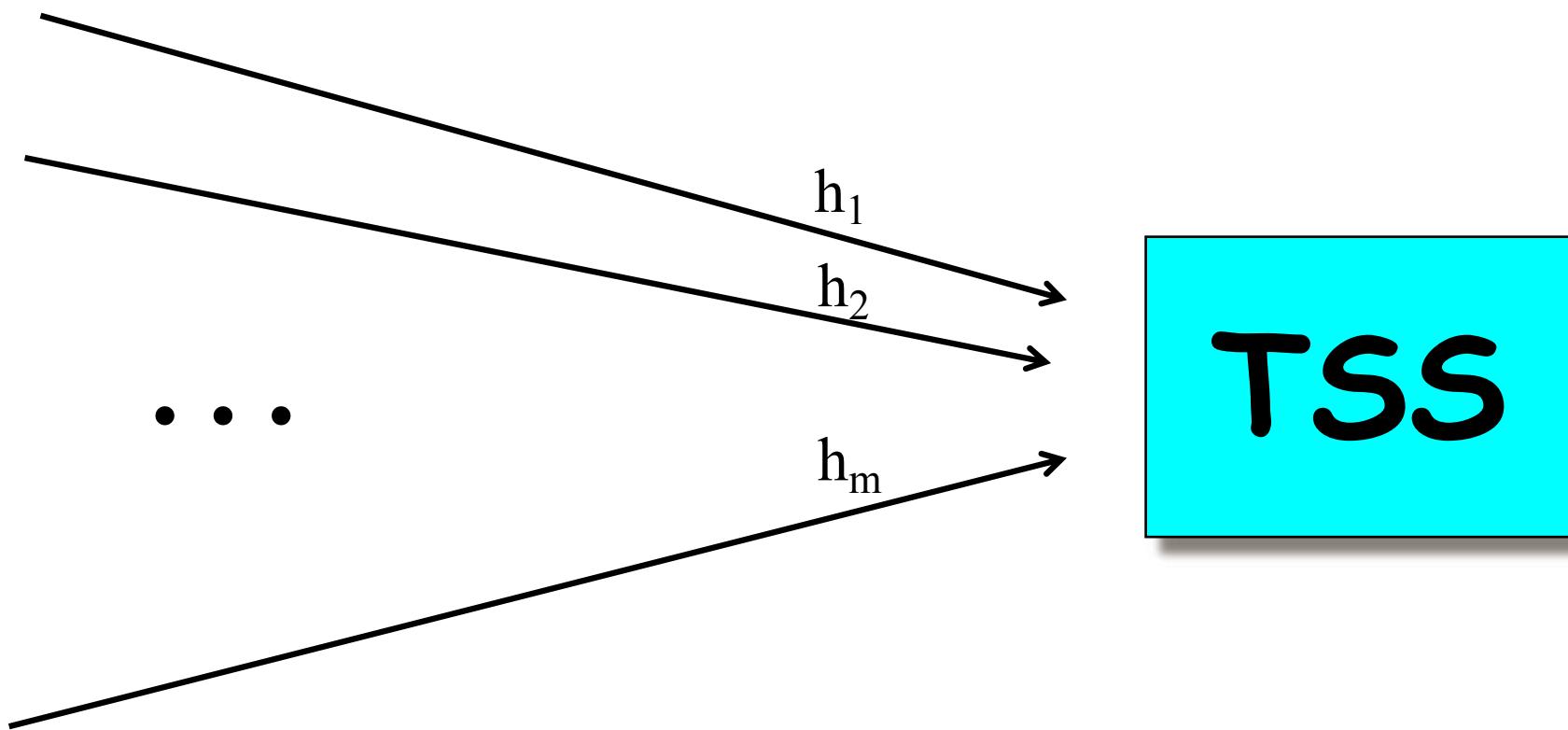
# Protocollo Distribuito: Problemi

- Ci vogliono molte persone in grado di rispondere immediatamente ad Alice
- Durata (vita) delle firme digitali:
  - La firma potrebbe non essere più valida al tempo della verifica della marca temporale:
    - La chiave privata è stata compromessa
    - Lo schema di firme è stato rotto

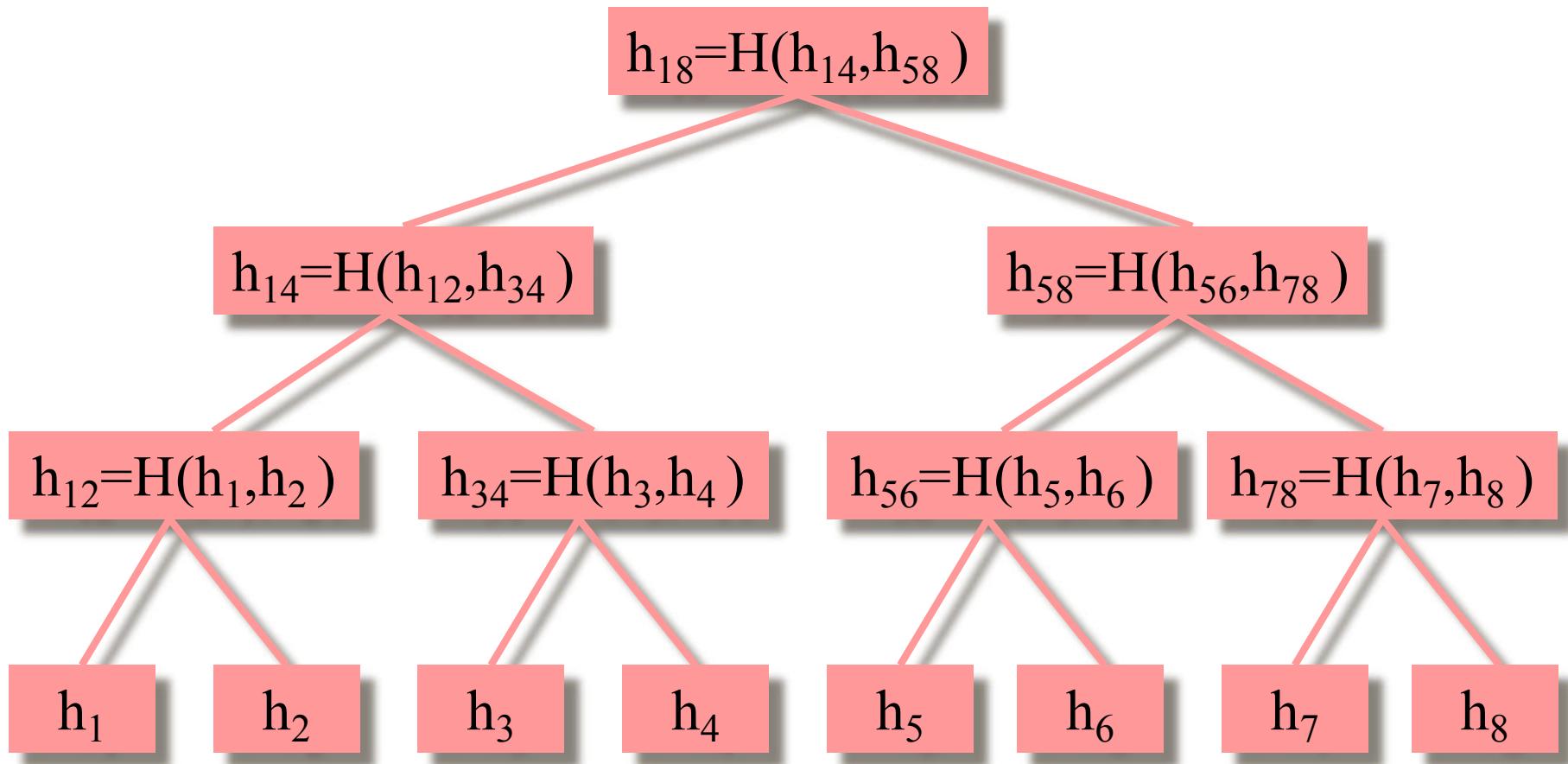
# Protocollo con “link”

- Time Stamping Service
  - (Autorità fidata, ... ma non troppo)
- Riceve tutte le richieste in intervalli prefissati
- Le collega tra loro
- Invia ad ognuno una marca temporale
- Vincola se stesso a “non poter predare”

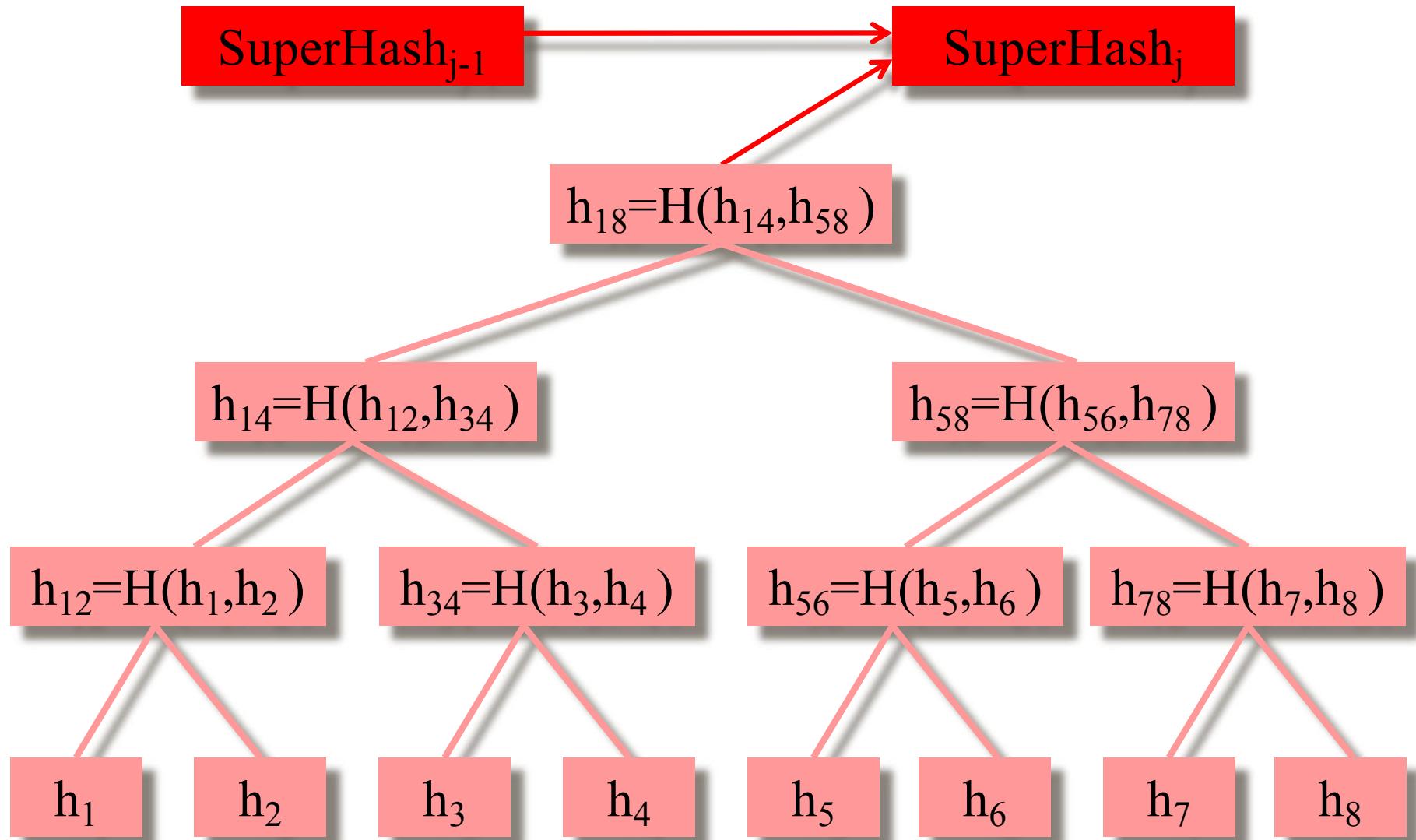
# Ricezione richieste in un prefissato intervallo di tempo



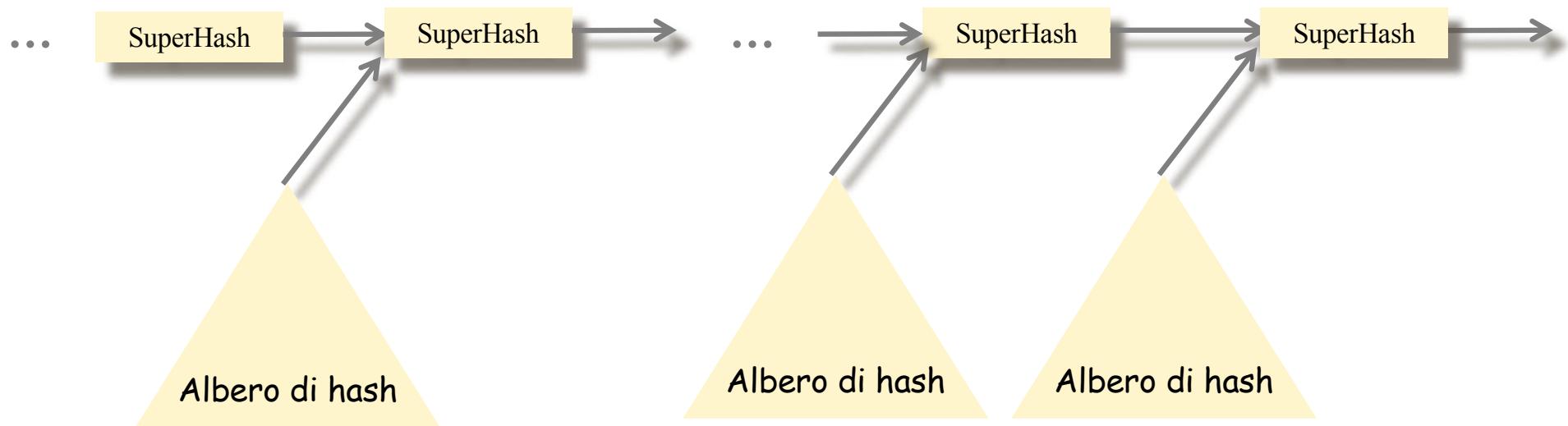
# Costruzione albero di hash



# Collegamento tra intervalli successivi



# Collegamenti SuperHash



# Marca temporale

Inviata per ogni richiesta ricevuta nell'unità di tempo

ID utente della richiesta

$h_i$

data ed ora

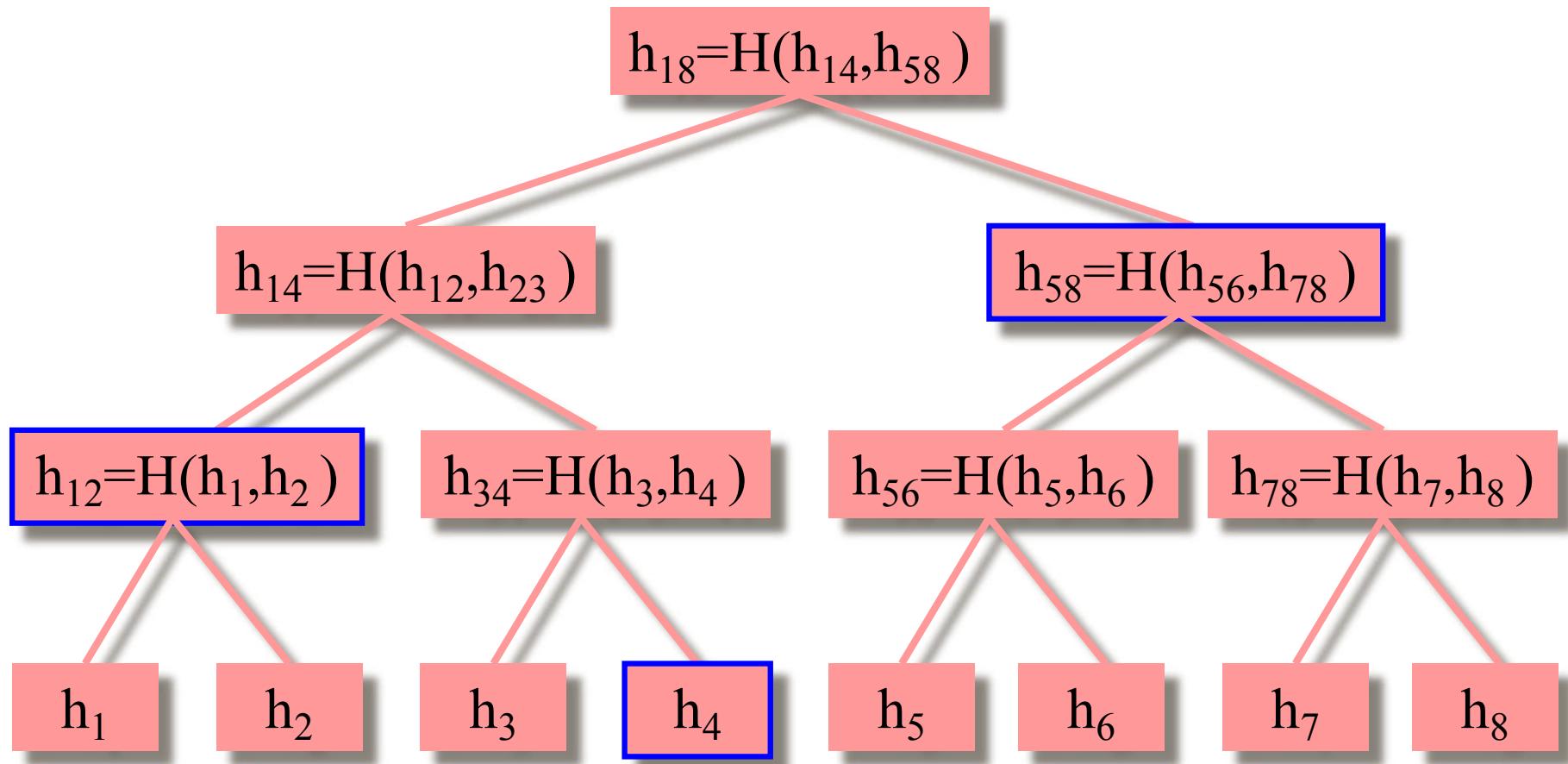
$h_{1m}$  (valore hash della radice dell'albero)

info necessarie per verificare che  $h_i$  è stato utilizzato per costruire l'albero con radice  $h_{1m}$

$\text{SuperHash}_{j-1}$  e  $\text{SuperHash}_j$

Firma del TSS

# Info per verifica di “ $h_3$ in albero con radice $h_{18}$ ”



# Sicurezza del Sistema

Fissato il valore hash della radice,  
non è possibile

- inserire un nuovo valore nell'albero di hash
- cambiare anche un solo valore nell'albero di hash

...altrimenti si determinerebbe una collisione  
per la funzione hash

# Sicurezza del Sistema

➤ Si potrebbe rompere lo schema colludendo solo con il TSS e creando una insieme di alberi collegati lunghi "a sufficienza"



➤ Una possibile soluzione:

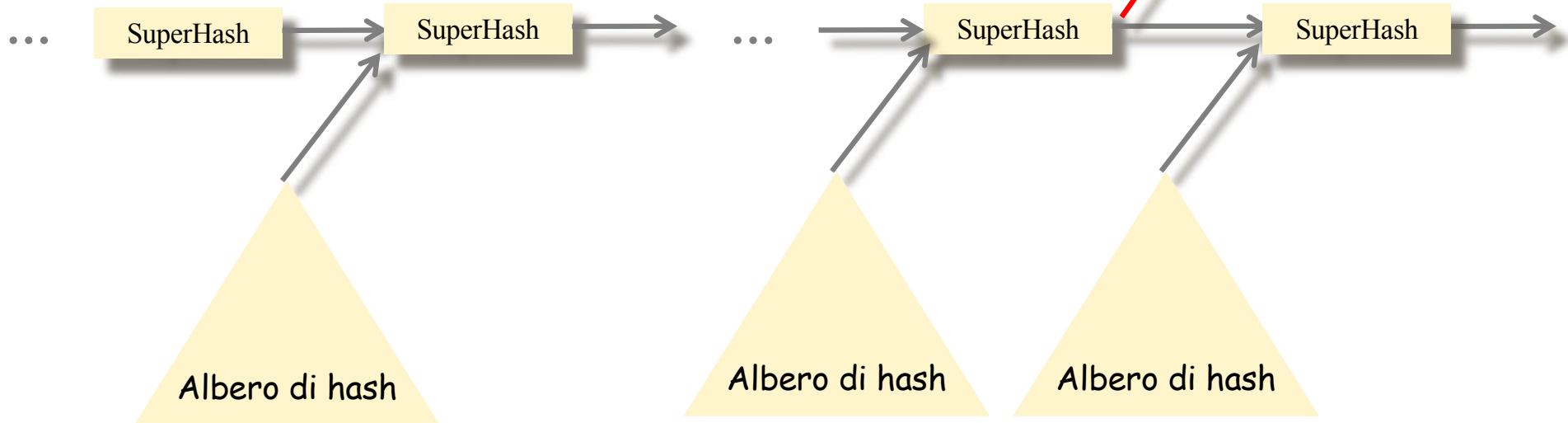
pubblicizzare SuperHash ad intervalli regolari

- ogni giorno su Internet, su quotidiani,...
- distribuzione mediante e-mail, CD,...





Pubblicizzazione



# Digital Notary

<http://www.surety.com>

- Il cliente usa del software venduto dalla Surety
- Funzione hash con un digest di 288 bit (MD5+SHA)
- Il sistema usa una struttura ad albero
- L'unità di tempo corrisponde ad un secondo
- Un numero seriale è inserito nel documento
- Il SuperHash è pubblicato in posti accessibili via rete, su un CD-ROM, ed ogni settimana sul Sunday New-York Times (nella sezione "Notices & Lost and Found" degli annunci a pagamento)

# La blockchain più antica del mondo si nasconde in un giornale dal 1995

È sempre stata lì sotto gli occhi di tutti nelle copie del New York Times dal 1995.

Di Daniel Oberhaus

28 agosto 2018, 3:23pm [Facebook](#) [Twitter](#) [Snap](#)



<https://www.vice.com/it/article/j5nzx4/blockchain-surety-new-york-times-1995-haber-stornetta>

# PGP Digital Timestamping Service

<http://www.itconsult.co.uk/stamper.htm>

Dal 12 ottobre 1995

- Il TSS firma ogni documento che riceve
- Ogni firma ha un numero seriale
- Il TSS memorizza tutte le firme che genera
- Tutte le marche (Serial Number, Date, Time) emesse possono essere esaminate

# PGP Digital Timestamping Service

<http://www.itconsult.co.uk/stamper.htm>

## Stamper Signature & Summary Files

---

The summary signatures from Stamper are available here as part of the public record of what signatures have been made. This should lend weight to the trustworthiness of the service. Full details about the information available here is contained within the [Stamper Information](#) document.

- [1995 Signatures by date](#)
- [1996 Signatures by date](#)
- [1997 Signatures by date](#)
- [1998 Signatures by date](#)
- [1999 Signatures by date](#)
- [2000 Signatures by date](#)
- [2001 Signatures by date](#)
- [2002 Signatures by date](#)
- [2003 Signatures by date](#)
- [2004 Signatures by date](#)
- [2005 Signatures by date](#)
- [2006 Signatures by date](#)
- [2007 Signatures by date](#)
- [2008 Signatures by date](#)
- [2009 Signatures by date](#)
- [2010 Signatures by date](#)
- [2011 Signatures by date](#)
- [2012 Signatures by date](#)
- [2013 Signatures by date](#)
- [2014 Signatures by date](#)
- [2015 Signatures by date](#)
- [2016 Signatures by date](#)
- [2017 Signatures by date](#)
- [2018 Signatures by date](#)
- [2019 Signatures by date](#)
- [2020 Signatures by date](#)
- [Weekly Summary Signatures](#)
- [Detached Signatures by Date](#)

# PGP Digital Timestamping Service

<http://www.itconsult.co.uk/stamper.htm>

## Stamper Signature & Summary Files

The summary signatures from Stamper are available here as part of the public record of what signatures have been made. This should lend weight to the trustworthiness of the service. Full details about the information available here is contained within the [Stamper Information](#) document.

- [1995 Signatures by date](#)
- [1996 Signatures by date](#)
- [1997 Signatures by date](#)
- [1998 Signatures by date](#)
- [1999 Signatures by date](#)
- [2000 Signatures by date](#)
- [2001 Signatures by date](#)
- [2002 Signatures by date](#)
- [2003 Signatures by date](#)
- [2004 Signatures by date](#)
- [2005 Signatures by date](#)
- [2006 Signatures by date](#)
- [2007 Signatures by date](#)
- [2008 Signatures by date](#)
- [2009 Signatures by date](#)
- [2010 Signatures by date](#)
- [2011 Signatures by date](#)
- [2012 Signatures by date](#)
- [2013 Signatures by date](#)
- [2014 Signatures by date](#)
- [2015 Signatures by date](#)
- [2016 Signatures by date](#)
- [2017 Signatures by date](#)
- [2018 Signatures by date](#)
- [2019 Signatures by date](#)
- [2020 Signatures by date](#)
- [Weekly Summary Signatures](#)
- [Detached Signatures by Date](#)

Ogni giorno pubblica

➤ Numero seriale dell'ultima firma effettuata

```
1140960 2020/05/06 00:01
1141219 2020/05/07 00:01
1141480 2020/05/08 00:01
1141735 2020/05/09 00:01

-----BEGIN PGP SIGNATURE-----
Version: 2.6.3i
Charset: noconv

iQCVAgUBXrXkrOD1+HYIOgipAQFuiwP+MUO+vKQavhMEMuiQLFZiWGt6zrtNpDtZ
las60XsX110K2kKDNOnnT/zm7dhB+1SWXa1Sld/I13/rMIZL/u6odlwnmjCWBcE9
oGthsUiQM3fvXxg99agzQWcf62NqLmGHkN+3YrobeYYilxL/TkrUypDELiGiqmHe
nRzPttQrSbw=
=/M8k
-----END PGP SIGNATURE-----
```

➤ Tutte le marche emesse nella giornata

	<a href="#">20200505.txt</a>	06-May-2020 03:55	103K
	<a href="#">20200506.txt</a>	07-May-2020 03:55	102K
	<a href="#">20200507.txt</a>	08-May-2020 03:55	103K
	<a href="#">20200508.txt</a>	09-May-2020 03:55	101K

# PGP Digital Timestamping Service

<http://www.itconsult.co.uk/stamper.htm>

## Stamper Signature & Summary Files

The summary signatures from Stamper are available here as part of the public record of what signatures have been made. This should lend weight to the trustworthiness of the service. Full details about the information available here is contained within the [Stamper Information](#) document.

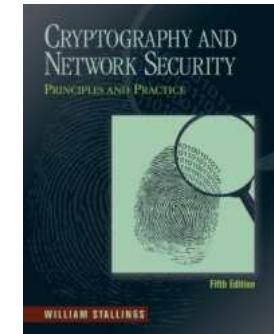
- [1995 Signatures by date](#)
- [1996 Signatures by date](#)
- [1997 Signatures by date](#)
- [1998 Signatures by date](#)
- [1999 Signatures by date](#)
- [2000 Signatures by date](#)
- [2001 Signatures by date](#)
- [2002 Signatures by date](#)
- [2003 Signatures by date](#)
- [2004 Signatures by date](#)
- [2005 Signatures by date](#)
- [2006 Signatures by date](#)
- [2007 Signatures by date](#)
- [2008 Signatures by date](#)
- [2009 Signatures by date](#)
- [2010 Signatures by date](#)
- [2011 Signatures by date](#)
- [2012 Signatures by date](#)
- [2013 Signatures by date](#)
- [2014 Signatures by date](#)
- [2015 Signatures by date](#)
- [2016 Signatures by date](#)
- [2017 Signatures by date](#)
- [2018 Signatures by date](#)
- [2019 Signatures by date](#)
- [2020 Signatures by date](#)
- [Weekly Summary Signatures](#)
- [Detached Signatures by Date](#)

Ogni settimana pubblica  
➤ Weekly summary signatures

 <a href="#">wk2020041.txt</a>	05-Apr-2020 11:00	5.5K
 <a href="#">wk2020042.txt</a>	12-Apr-2020 09:18	5.5K
 <a href="#">wk2020043.txt</a>	19-Apr-2020 08:52	5.5K
 <a href="#">wk2020044.txt</a>	26-Apr-2020 11:14	5.5K
 <a href="#">wk2020051.txt</a>	03-May-2020 08:38	5.5K

# Bibliografia

- **Cryptography and Network Security**  
by W. Stallings, 2010
  - cap. 11 + Appendice 11.A, cap. 12 (SHA)
- Tesina di Sicurezza su reti
  - Funzioni hash
- The Keccak reference
  - <http://keccak.noekeon.org/Keccak-reference-3.0.pdf>



# Domande?

