

3. Crittografia

Sicurezza dell'Informazione

Avvertimento

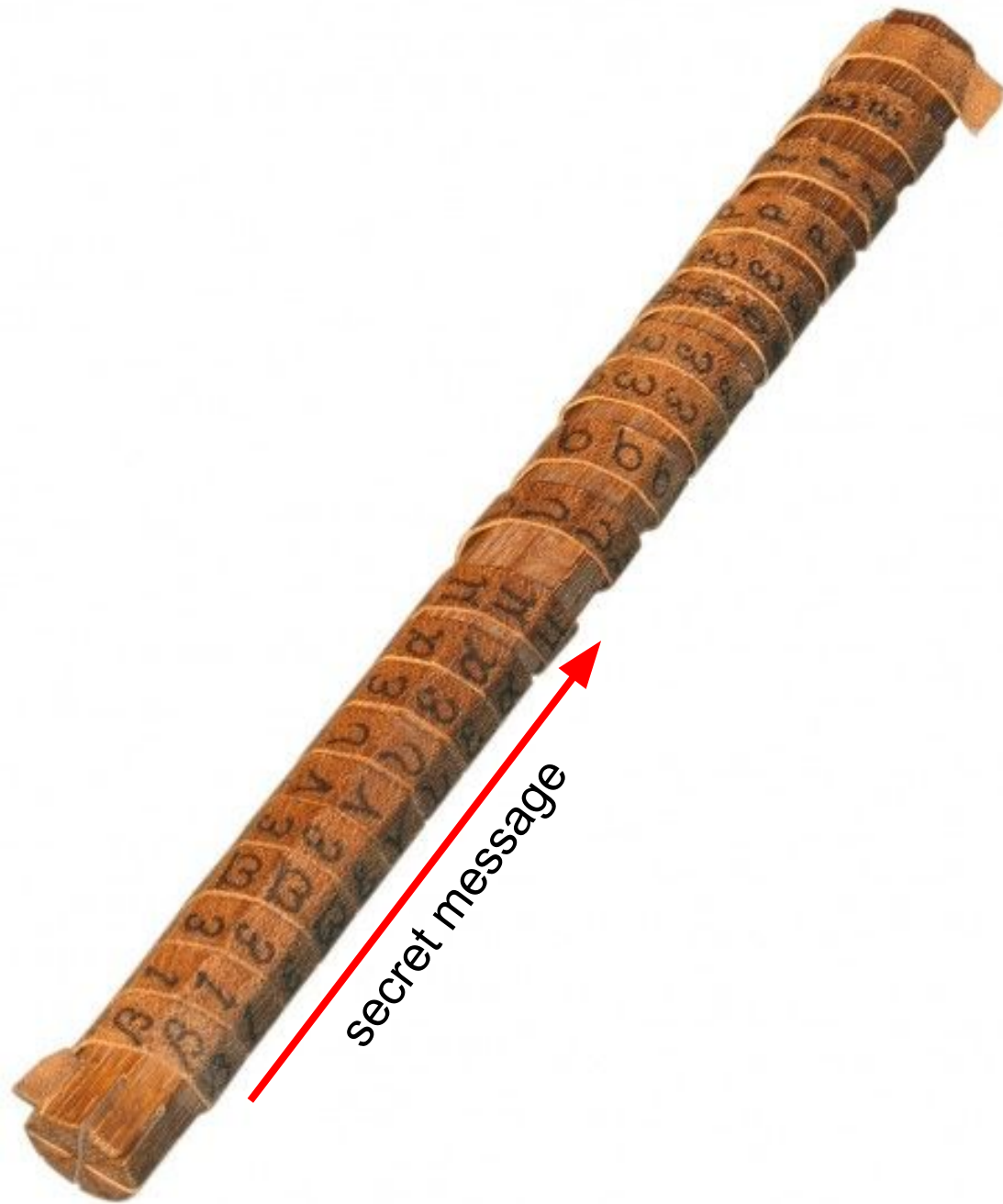
- Questa è una breve introduzione semplificata alla crittografia.
- Presenteremo solo ciò che è necessario per le discussioni sulla sicurezza dei sistemi.
- Nella maggior parte dei casi tratteremo i concetti matematici come delle “scatole nere”.

Un cenno storico

Dal greco: *kryptós* = “nascosto” e *gráphein* = “scrivere” (ossia “l’arte della scrittura segreta”).

Nell’antichità, la scrittura stessa era già una “tecnica segreta”.

La crittografia nasce nella società greca, quando la scrittura divenne più comune e si sentì il bisogno di sviluppare forme di scrittura nascosta.



Il cifrario di Cesare (o a rotazione)

m

C	R	Y	P	T	O	G	R	A	P	H	Y
2	17	24	15	19	14	6	17	0	15	7	24




chiave $k = 5$

c

7	22	3	20	24	19	11	22	5	20	12	3
H	W	D	U	Y	T	L	W	F	U	M	D

Cifrario a sostituzione

\mathcal{A}_m	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
\mathcal{A}_c	t	m	k	g	o	y	d	s	i	p	e	l	u	a	v	c	r	j	w	x	z	n	h	b	q	f



m C R Y P T O G R A P H Y



c k j q c x v d j t c s q

Pigpen

A	B	C
D	E	F
G	H	I

J .	K .	L .
M .	N .	O .
P .	Q .	R .

S
T **U**
V

W
X **Y**
Z

Cifrario di Vigenère (multi-rotazione)

m

C	R	Y	P	T	O	G	R	A	P	H	Y
2	17	24	15	19	14	6	17	0	15	7	24

chiave $k = \{5, 12, 7\}$



c

5	3	5	18	5	21	9	3	7	18	19	5
F	D	F	S	F	V	J	D	H	S	T	F

Kasisky Test

"OTFDWP DZ DJDG IMKDHIG ASSUKOE, NQYBZZVBI ZZRZWYVBXP NDPVZTNI JZMLCO, IN UW RCXVDKRDO NADN IXZOZJB DDDKDU WA UKB EEI, MJ RD UVDSUUA, EMGXDQXMJ, JDL KQRABA; VZU CXWPSY EQZDALB DIOUFWI PVHV RD AJYV FQG MQTXWVDLVM Q KZDKJYV MAPJB XMQVVYVZZTN YV OTZB XCZ; QMND BCQ SJHJVDZL, WZVZU XBL FUEPI WA BVPK XGMTRDO OTV CYBGQ "CXHL JR **KQU** ECUKN UTZBYJDBN" MSXLM VXC CXMDD FCXMM YRPDQGAHDUVO MJLHQKFZXDA JR UXCQIUFW; QVY **FYN** CWYQIW AQISJ XV ADMD DDNPDCRDO OTV BQUZ EEXM-ECUKN GCVPIDFMY UE CXM MAPJB AOMEMQZY; MEM **JPZ** TRWEDZDZJD NGMX KUIMUEP **JPZ** AEN VQBGIN EN V EEXM-ECUKN SPVDXNH; IIP **KQU** OMQRC QCNFIRQV ZYGRHM, XMVBQZDME, QUQM FF XLMMXFATQIS IXCM, CMMRDO AAI CXM DYGNHQVX TXBWM **FYN** IIHQ ZVFMMURU XCZ; MEM JPJGXQ JPDE GAU-MHUENDKZ UE RJ IKBCRUA OA **KQU** PPYRW HIXQ ZCIMGR, XRLQIS **KQU** ECUKN CII UUNQT HMJCUZNTZY EDZD VEUZT PLBAG ODZKU; IIP KQECBT, SNIQYQJ JBT OTZB, MPDFVWUAN TRB RMZZ VEUV HMUN IQBZZOYKVZK XV OGMUWUAN, RFA QUIZX CXM MADJDA V IYRJM NFFWU UVDBNT I EAPOKT YMP; JDL OTFDWP DZ FCXMM YFAJIG EPVFIOTZNI IIP JHCJJXZIYVBE, KQYA NMDN XCZ UJ VQLZ **FYN** UUWXVV EN HMEH JWPOYRDO, IASUU BCUEPI- BCQ ZWDWXQELU WA NIRTMN, **FYN** RMIUXWYBT AW JWM; OTFDWP VYFWW BCQ INT UZZ FO QUZDZLQ BCQ XRLQIS FO **JPZ** IYRJM WQCC EN RMDYKU RMJ CXM YQVYUAO BCNTOZ AW QEVDJ; KQECBT ZW CIIK TUYUZE, NQYBZZVBI BTBZOYMN **FYN** CIEQJCO WA VLBQXQ ZW **JPZ** QIVYVZ AW CXM EGUPU, IIP TXDBMUSDJMN FF CXM YMZUO AOMKN EN FUEPI IIP HDUMIE UAQEI NP VYTF-IYRJM NFNNTA"

Rivelava la lunghezza della chiave

Quando la matematica ha vinto la guerra

- Durante la Seconda Guerra Mondiale, Alan Turing lavorò a Bletchley Park per decifrare i codici delle potenze dell'Asse, in particolare il cifrario Enigma.
- La nascita dei primi computer universali fu in parte stimolata proprio da questo sforzo di decrittazione.



Definizione

È la pratica e lo studio delle tecniche che forniscono:

- **Confidenzialità:** solo le parti autorizzate possono accedere alle informazioni.
- **Integrità:** i dati non possono essere alterati o modificati senza essere rilevato.
- **Autenticazione:** verifica dell'identità delle parti coinvolte nella comunicazione.
- **Non ripudio:** impedisce a una parte di negare di aver inviato o ricevuto un messaggio.

Cosa ci serve

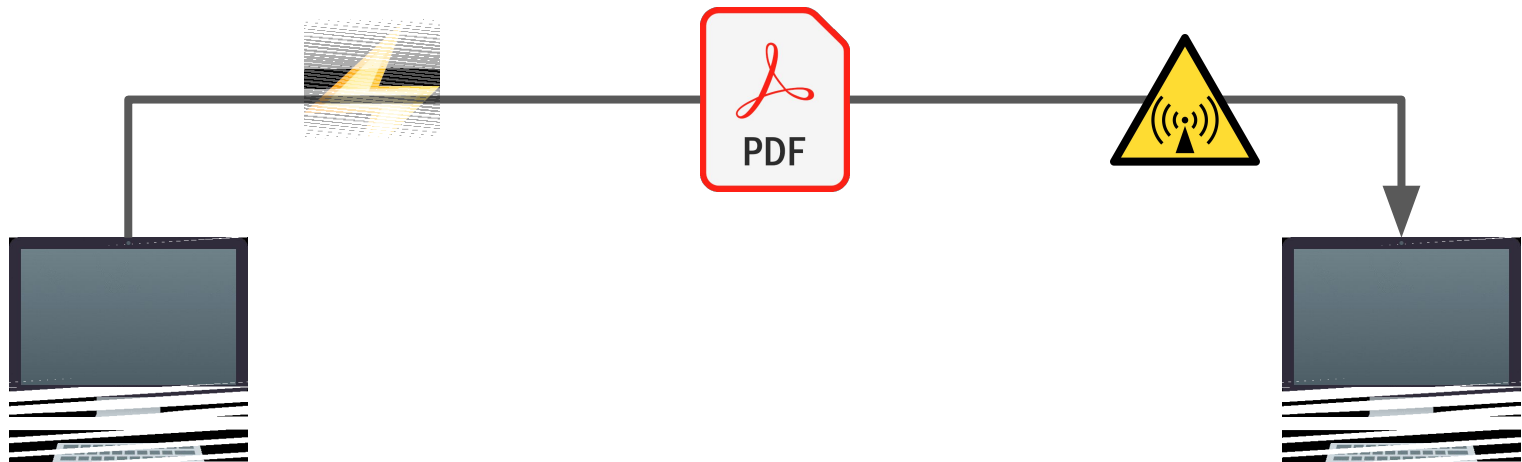
A noi interessa principalmente:

- **Funzioni di Hash:** per l'integrità dei dati.
- **Cifrari**
 - **simmetrici:** per comunicare velocemente con una sola chiave.
 - **asimmetrici:** per lo scambio di chiavi simmetriche e l'autenticazione.

Funzioni di Hash

Esempio: invio di file

Rumori di diversa natura possono causare alterazioni nei file che inviamo. Come ce ne accorgiamo?



Esempio: acquisizione di una prova digitale

Le fonti di prova (Hard disk e SSD) passano tramite diverse persone prima di arrivare in un tribunale:

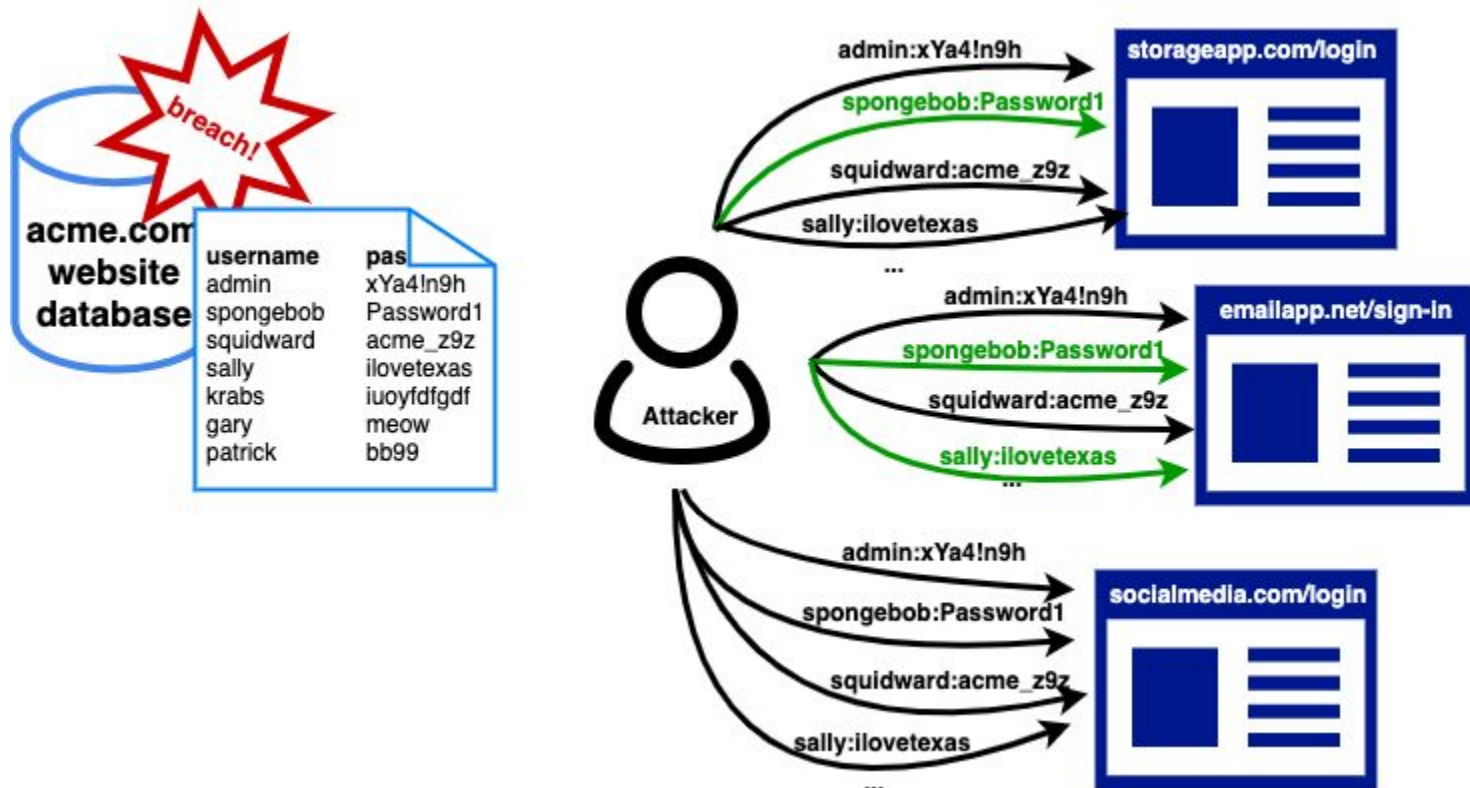
- Periti tecnici
- Supervisore delle indagini
- Analisti forensi
- Responsabili della custodia



Come sappiamo se è stata alterata?

Esempio: attacco informatico

Un hacker malevolo ruba le password da un sito. Molti utenti ri-usano le stesse password su altri siti. Possiamo anonimizzarle?



Idea generale

Ci serve uno strumento in grado di **verificare se qualcosa è cambiato**. Le funzioni di hash fanno questo.



*Dato un'entità digitale (binaria), la **funzione di hash** restituisce un numero (chiamato **digest**), che rappresenta quella specifica entità.*

Proprietà

Una funzione di hash deve essere:

- **Deterministica**: dato un input, restituisce sempre lo stesso digest.
- **Produrre un digest di lunghezza fissa (in bit)**: altrimenti risulterebbe complesso gestire diverse lunghezze.
- **Irreversibile**: dal digest non si deve poter risalire all'input.
- **Resistente alle collisioni**: deve essere difficile trovare 2 input con lo stesso digest.
- **Veloce da calcolare**: per ragioni di performance.

Una semplice funzione di hash

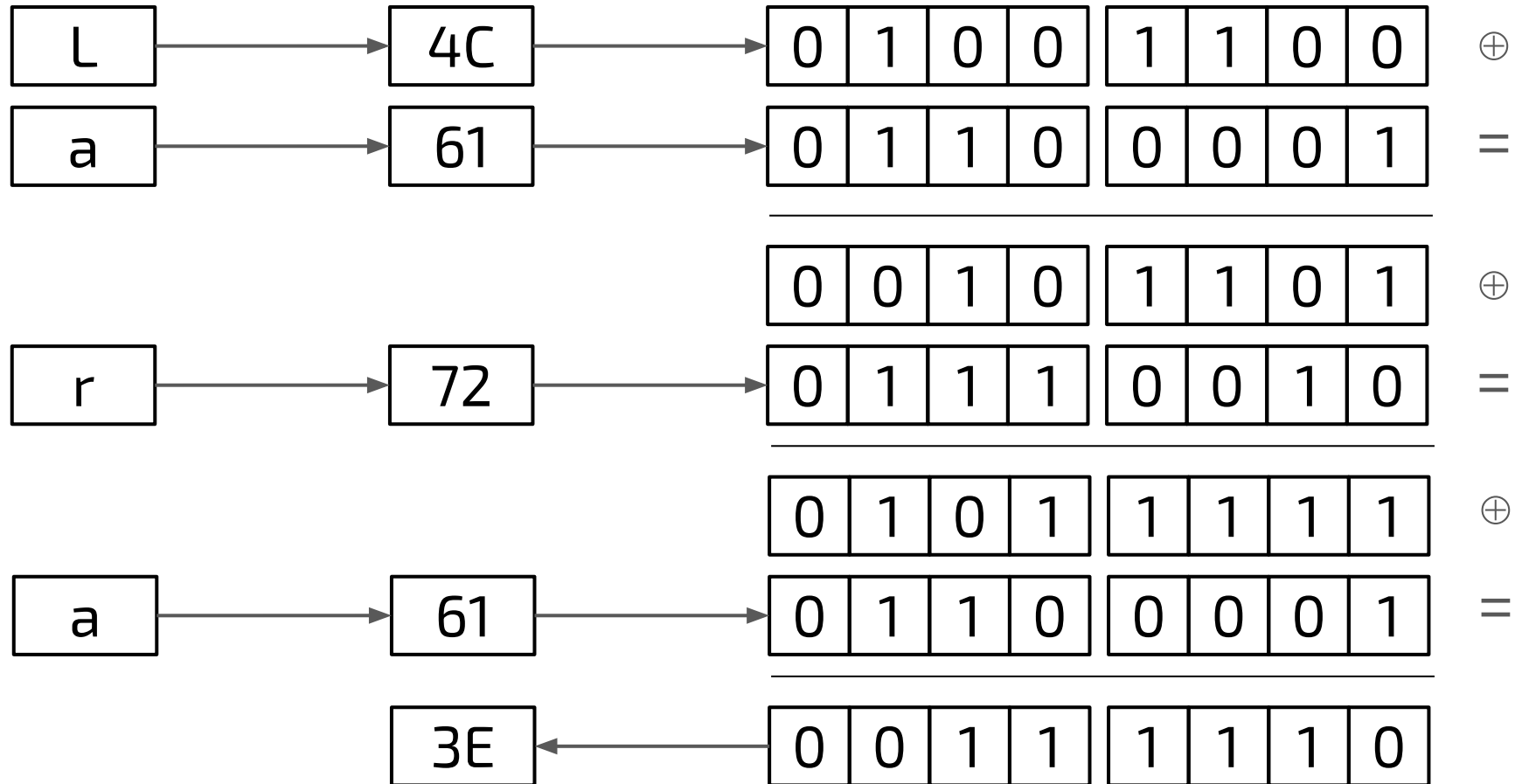
Un esempio di una funzione di hash è una funzione che prende tutti i byte e fa una xor fra di loro:

$$H(x) = x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n$$

Esempio:

L	a	r	a
4C	61	72	61

Una semplice funzione di hash



Una semplice funzione di hash

Digest di “Lara” è 0x3E.

E quello di “Enzo”?

Una semplice funzione di hash

Digest di “Lara” è 0x3E.

E quello di “Enzo”?

Questa funzione di hash è:

- **Deterministica**
- **Produce un digest di lunghezza fissa (in bit)**
- **Irreversibile**
- ~~**Resistente alle collisioni**~~
- **Veloce da calcolare**

Resistenza alle collisioni

L'aspetto **più importante** di una funzione di hash è la **resistenza alle collisioni**. Senza questa caratteristica sarebbe facile:

- Non accorgersi di un cambiamento dovuto al rumore
- Manomettere le fonti di prova
- Proteggere le informazioni online

Funzioni di hash moderne

Algoritmo	Lunghezza (bit)	Digest di "ciao"
MD5	128 ($2^{128} \approx 3,4 \times 10^{38}$)	6e6bc4e49dd477ebc98ef4046c067b5f
SHA1	160 ($2^{160} \approx 1,46 \times 10^{48}$)	1e4e888ac66f8dd41e00c5a7ac36a32a9950d271
SHA256	256 ($2^{256} \approx 1,16 \times 10^{77}$)	b133a0c0e9bee3be20163d2ad31d6248db292aa6d cb1ee2d7fc0da29886a2a5d
SHA512	512 ($2^{512} \approx 1,34 \times 10^{154}$)	a0c299b71a9e59d5ebb07917e70601a3570aa103e 99a7b2c92e4b0fef2d8a6a26e2d64cd845c7f0fbc 7b383e4ac2a32d7f49b2911b0a09e301b78a35fd9 d69fc

Resistenza alle collisioni

Per resistenza alle collisioni si intende che è difficile trovare due input **qualsiasi** con lo stesso digest.

Casa	634e888a
Cucina	0e9ba298
Mobile	a2a86fef
Garage	d7231f49
Radio	0e9ba298
Microfono	d64cd845

Resistenza alla prima pre-immagine

Dato un digest ***d***, deve essere difficile trovare un input ***i*** tale che **$H(i) = h$** .

Digest: 1e4e888a

Deve essere difficile trovare un input ***i***, per esempio “segreto”, che da un digest ***d*** pari a 1e4e888a

Resistenza alla seconda pre-immagine

Dato un input i_1 , deve essere difficile trovare un input diversi i_2 tale che $H(i_1) = H(i_2)$.

Input: "Pippo" -> **Digest:** 4e478089

Deve essere difficile trovare un input i_2 che da lo stesso digest **d** di "Pippo" (ovvero 4e478089 nell'esempio)

Resistenza e tempi medi

Algoritmo	Collisione	Pre-Immagine
MD5	secondi	10^{18} anni
SHA1	giorni	10^{28} anni
SHA256	10^{32} anni	10^{57} anni
SHA512	10^{66} anni	10^{134} anni

Resistenza e tempi medi

Algoritmo	Collisione	Pre-Immagine
MD5	secondi	10^{18} anni
SHA1	giorni	10^{28} anni
SHA256	10^{32} anni	10^{57} anni
SHA512	10^{66} anni	10^{134} anni

MD5 e SHA1 sono oggi considerati insicuri

GDPR: Password e sicurezza online

Articolo 5, par. 1, lett. f) — Integrità e riservatezza

I dati personali devono essere trattati in modo da garantire un'adeguata sicurezza, compresa la protezione contro il trattamento non autorizzato o illecito e contro la perdita, la distruzione o il danno accidentale, mediante misure tecniche e organizzative adeguate.

GDPR: Password e sicurezza online

Se si vuole essere compliant con il GDPR, occorre proteggere/anonimizzare le password. L'hashing è la soluzione.

Username	Password Hash
Law	1e4e888a
Frank	0e9bee3b
Matteo	2e4b0fef
Rob	32d7f49b
Gab	e2d7fc0d
Tom	d64cd845

GDPR: Password e sicurezza online

Se l'attaccante riesce ad accedere al database le nostre password non sono del tutte compromesse. A meno che...

Username	Password Hash
Law	1e4e888a
Frank	0e9bee3b
Matteo	2e4b0fef
Rob	32d7f49b
Gab	e2d7fc0d
Tom	d64cd845

GDPR: Password e sicurezza online

Lookup Table

Username	Password Hash
Law	1e4e888a
Frank	0e9bee3b
Matteo	2e4b0fef
Rob	32d7f49b
Gab	e2d7fc0d
Tom	d64cd845

... l'attaccante non genera i digest di un dizionario di parole.
Nel gergo **Lookup Table**.

segreto	634e888a
password	1e4e888a
123456	a2a86fef
asdasd	d7231f49
mamma	0e9ba298
lorenzo	d64cd845
2004	e2d7fc0d
viadei...	683dea23
capricorno	54fe1082
rocky	232aedf2
milan	12543dee

Soluzione: salting

Alle password vengono concatenati dei caratteri casuali e poi viene calcolato l'hash.

Username	Salt	Password Hash
Law	3x31LwpZ	1e4e888a
Frank	2wV9q4w8	0e9bee3b
Matteo	iJ9xG2W8	2e4b0fef
Rob	l6T8u189	32d7f49b
Gab	94H0L1tV	e2d7fc0d
Tom	3m8HytT5	d64cd845

Soluzione: salting

L'attaccante deve creare una Lookup Table per utente -> **Impraticabile**

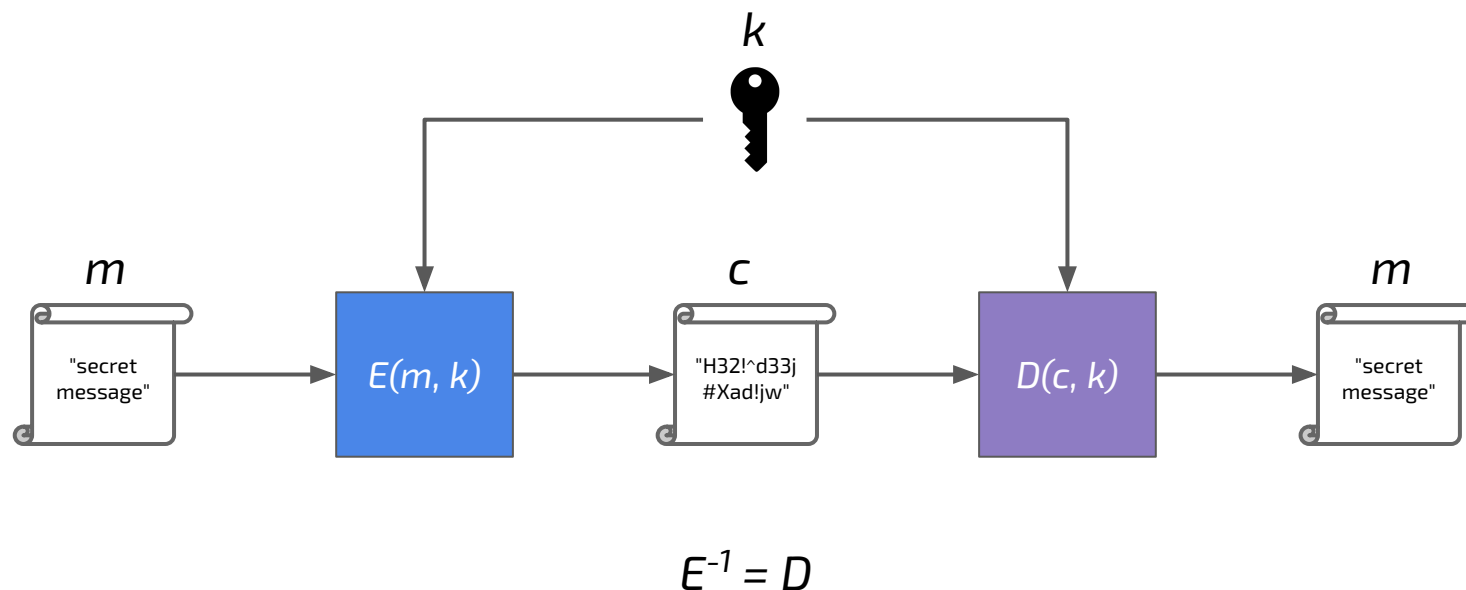
Username	Salt	Password Hash
Law	3x31LwpZ	2e4b0fef
Frank	2wV9q4w8	0e9bee3b
Matteo	iJ9xG2W8	32d7f49b
Rob	16T8u189	34fe7324
Gab	94H0L1tV	e2d7fc0d
Tom	3m8HytT5	d64cd845

Cifrari

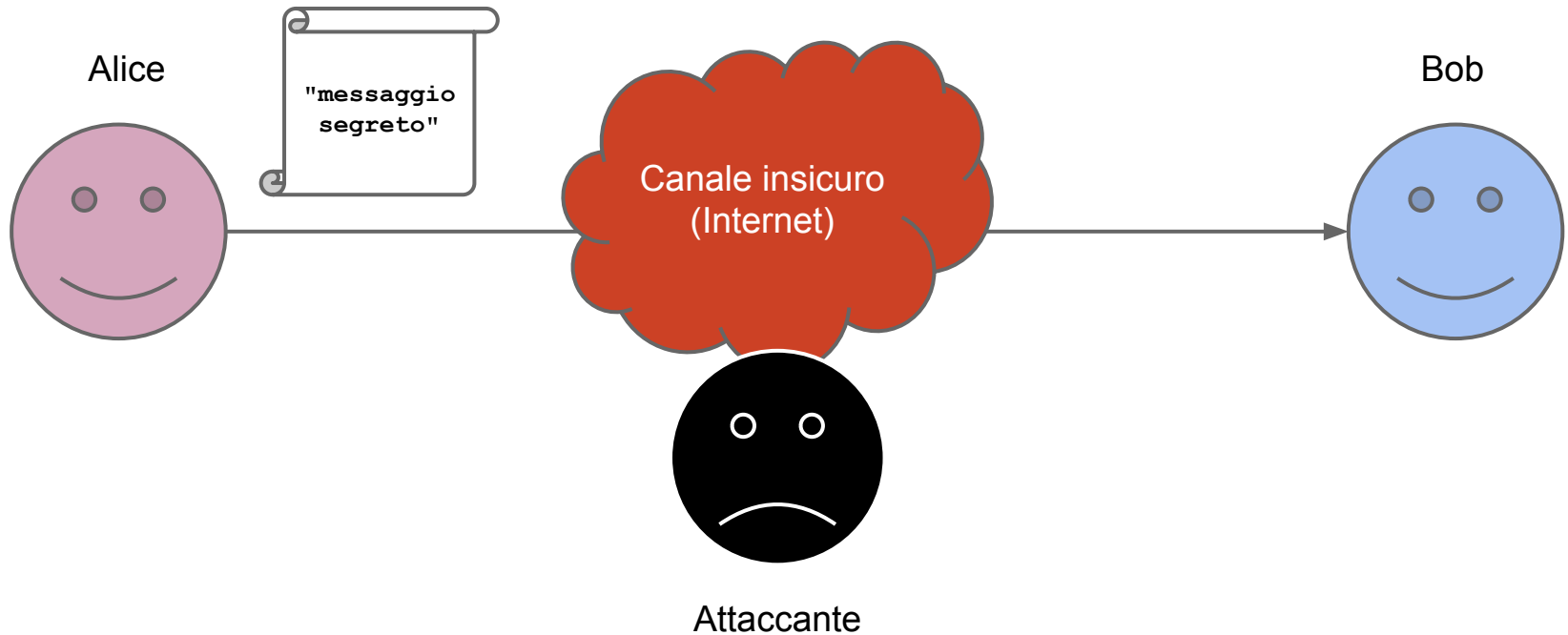
Crittosistema

Un crittosistema è un sistema composto da algoritmi crittografici.

Quando il suo obiettivo è garantire la riservatezza, esso prende in ingresso un messaggio (detto testo in chiaro) e lo trasforma in un testo cifrato mediante una funzione reversibile e una chiave.



Il problema da risolvere: confidenzialità



Il principio di Kerckhoffs

La sicurezza di un sistema crittografico deve basarsi solo sulla segretezza della chiave, e mai sulla segretezza dell'algoritmo.

Auguste Kerckhoffs, "La cryptographie militaire", 1883

Questo significa che:

- In un sistema crittografico sicuro non è possibile ricavare il testo in chiaro dal testo cifrato senza conoscere la chiave.
- Inoltre, non è possibile determinare la chiave analizzando coppie di testo in chiaro e testo cifrato.
- Gli algoritmi devono sempre essere considerati noti all'attaccante.

Il teorema di Shannon (1949)

Shannon definisce una cifratura perfetta come un sistema in cui:

conoscere il testo cifrato non dà alcuna informazione sul testo in chiaro.

In un cifrario perfetto, il numero di chiavi $|K|$ deve essere maggiore o uguale al numero di messaggi possibili $|M|$

$$|K| \geq |M|$$

Osservazione: se mando due volte lo stesso messaggio con la stessa chiave, rivelo una informazione.

Il cifrario perfetto: One-Time Pad (OTP)

- XOR di un messaggio m con una chiave casuale k della stessa lunghezza di m :

$$\text{lunghezza}(k) = \text{lunghezza}(m)$$

- La chiave è pre-condivisa e viene consumata durante la scrittura.
 - Non può mai essere riutilizzata!
- L'OTP (One-Time Pad) è un cifrario perfetto minimale:
 - minimale perché $|K| = |M|$
- ma terribilmente scomodo, usato solo in contesti speciali (es. comunicazioni diplomatiche o militari ad alta sicurezza).