Introduzione alla Crittografia

sulla sicurezza ed altre amenità informatiche

Mirko Zichichi · mirko.zichichi@iota.org

slide in collaborazione con prof. Stefano Ferretti

Seminario "Blockchain e Smart Contract: aspetti giuridici e tecnologici"
Università di Bologna, novembre 2026

Chi sono

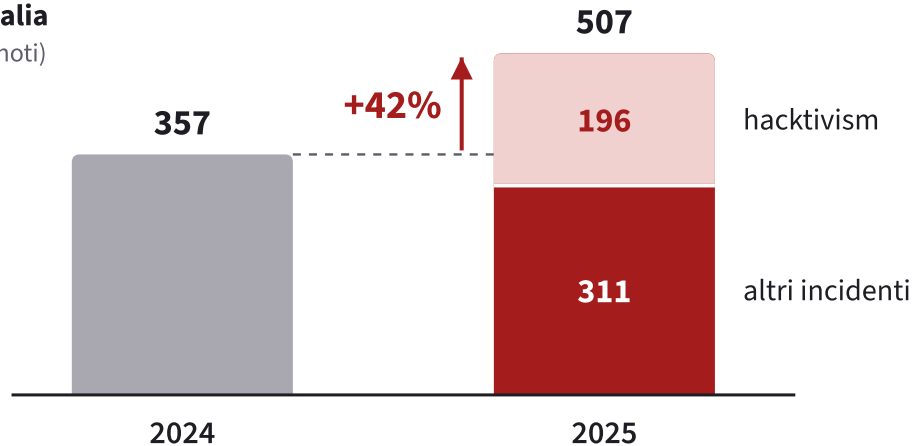
Mirko Zichichi, mirkozichichi.me

- **Engineering Team Lead** alla **IOTA Foundation**: registri distribuiti e smart contract (linguaggio Move)
- **Dottorato in Law, Science and Technology** (Bologna, Torino, Madrid)
- Tesi: *sistemi decentralizzati per proteggere i dati personali*

Un informatico tra i giuristi: oggi traduciamo la crittografia in concetti che già usate: firma, data certa, sigillo, busta chiusa.

L’Italia nel mirino: i numeri

Incidenti gravi in Italia
fonte: Clusit (incidenti noti)



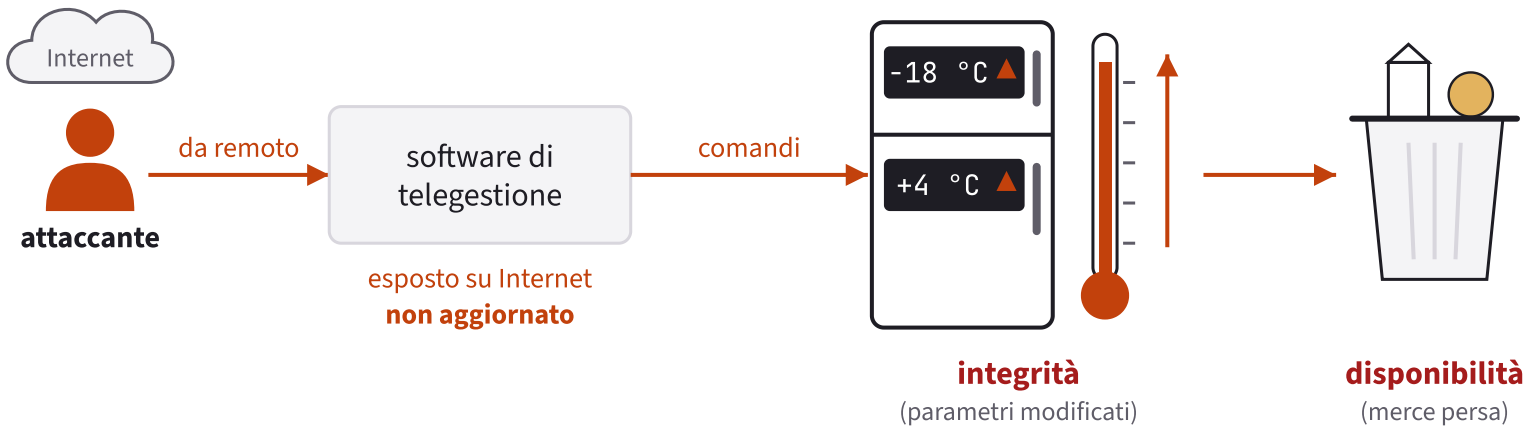
senza hacktivism
+12%
(in entrambi gli anni)

E il 2026? Nei primi nove mesi il CSIRT Italia dell'ACN ha gestito più eventi che in tutto il 2025 (2.729, di cui 615 incidenti).

Rapporto Clusit 2026 (17 marzo 2026); Relazione annuale ACN 2025; Operational Summary ACN 2026.

2 novembre 2026: i frigoriferi dei supermercati

Alle 4 del mattino qualcuno alza da remoto la temperatura dei frigoriferi in oltre 20 supermercati, da Palermo a Trapani.



In Europa oltre 100 supermercati colpiti; la rivendicazione filorussa è **contestata** da esperti e ACN.

Il Post, 3 e 6 novembre 2026; bollettino ACN per la filiera del freddo.

Un anno di attacchi: proviamo a classificarli

Per ogni notizia: quale proprietà della sicurezza è stata violata? Cliccate su una scheda.

2 nov 2026 — ✓ confermato

Frigoriferi manomessi nei supermercati, da Palermo a…

Temperature alzate da remoto in oltre 20 punti vendita in Sicilia e in altre regioni, sfruttand…

Violate: **integrità** **disponibilità**

art. 14 Cyber Resilience Act — Il Post, 3/11/2026

8 ott 2026 — ✓ confermato

Attacco al sito della Farnesina

Un attacco DDoS, attribuito dalla stampa al gruppo filorusso NoName057(16), viene…

Violate: **disponibilità**

d.lgs. 138/2024 (NIS2): notifica entro 24 ore

4 feb 2026 — ✓ confermato

Milano-Cortina: attacchi ai siti dei Giochi

Il governo dichiara sventati attacchi di origine russa a sedi diplomatiche e a siti legati alle…

Violate: **disponibilità**

L. 90/2024 sulla cybersicurezza nazionale — T

5 mar 2026 — ✓ confermato

Spyware Paragon: le procure confermano

Le procure di Roma e Napoli trovano tracce di malware sui telefoni di due attivisti e del…

Violate: **riservatezza** **integrità**

art. 615-ter c.p. (accesso abusivo) — Internazi…

feb 2025 — ✓ confermato

La voce clonata del ministro Crosetto

Con una voce ricreata al computer, truffatori ottengono da un imprenditore circa 980.000 …

Violate: **autenticità**

artt. 640 e 494 c.p. (truffa, sostituzione di perso

9 nov 2026 — *solo rivendicato*

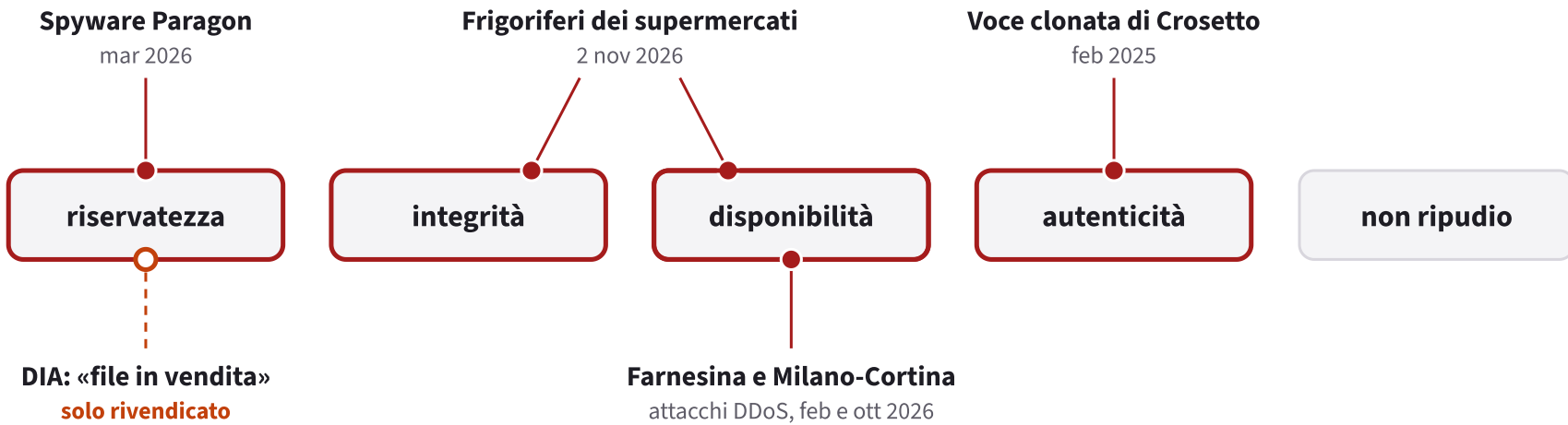
File della DIA in vendita sul dark web?

Un venditore offre 302 file attribuiti alla Direzione investigativa antimafia. Per gli…

Violate: **riservatezza**

una rivendicazione non è una prova — AGI, …

Sei notizie, cinque proprietà



Il percorso di oggi

I Sicurezza informatica

II Crittografia

III Crittografia simmetrica

IV Crittografia asimmetrica

V Funzioni hash

VI Puntatori hash e strutture dati

VII Non ripudio e firma digitale

VIII Marcatura temporale

IX Identità decentralizzate

I

Capo I

Sicurezza informatica

Cosa vogliamo proteggere, da chi, e perché non basta mai

La sicurezza informatica

Proteggere le risorse informatiche da accessi, usi, alterazioni o distruzioni non autorizzati.

Sicurezza fisica

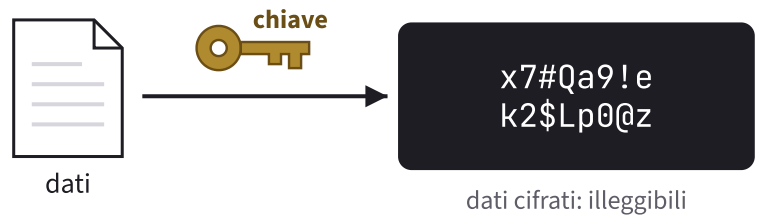
protegge i dispositivi e i locali



porte blindate, allarmi, casseforti

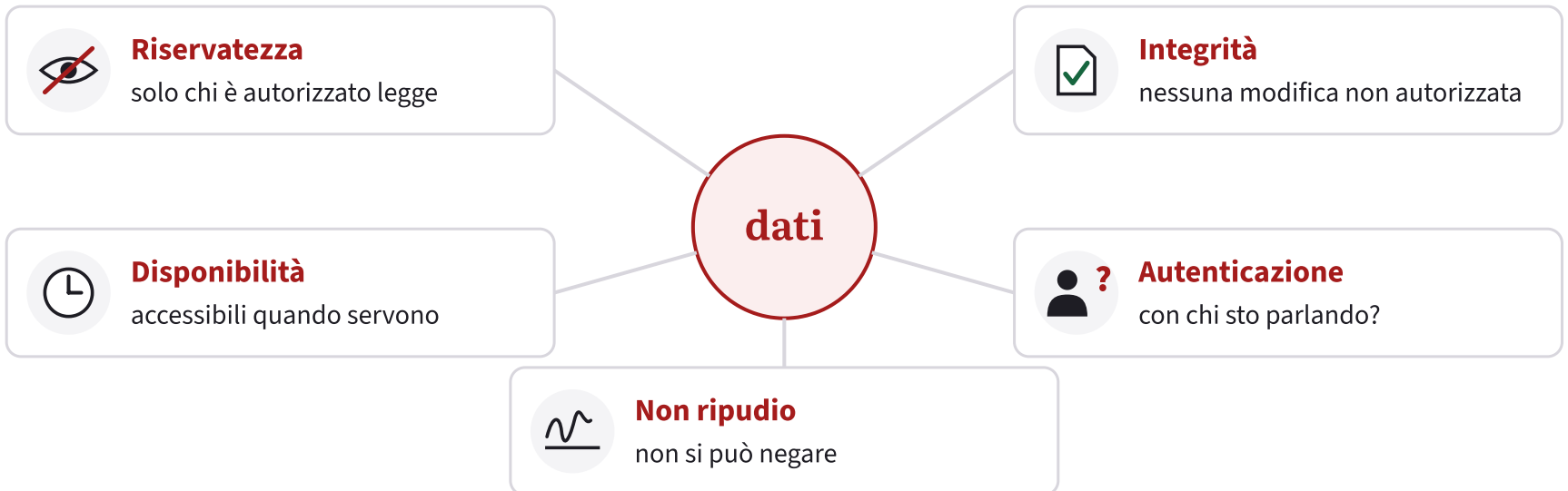
Sicurezza logica

protegge le informazioni



crittografia, password, firma elettronica

Sicurezza informatica: la terminologia



Tre aspetti fondamentali

1

Non esistono sistemi sicuri

2

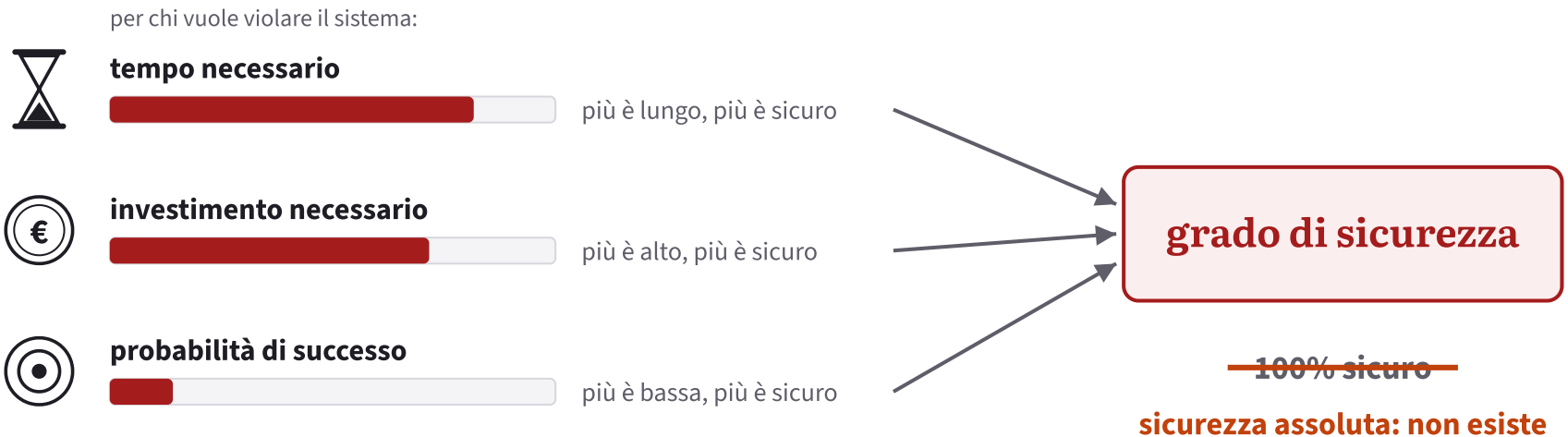
Sicurezza = conoscenza

3

Hardware, software e persone

1. Non esistono sistemi sicuri

- Il software non è mai perfetto; anche i protocolli possono avere errori di progetto
- Il sistema inviolabile è un mito, come il caveau impenetrabile o la nave inaffondabile (il Titanic)



1. Più complesso, più insicuro

sistema semplice

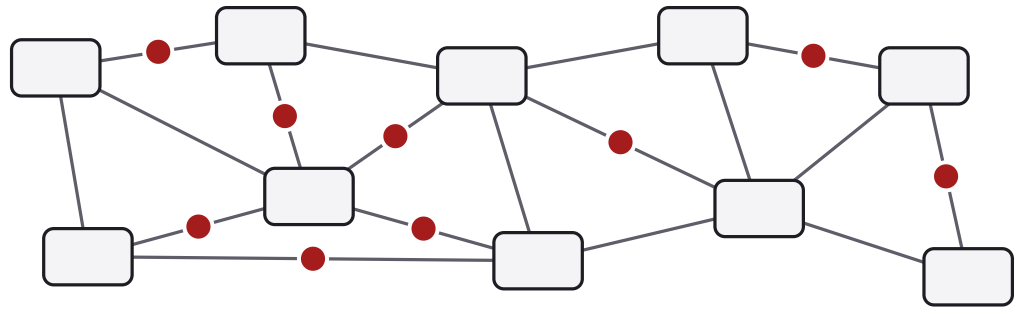
3 componenti



possibile punto debole

sistema complesso

10 componenti, 18 collegamenti: molti più punti deboli



«Quello che non c'è non si rompe»

Regola **KISS**: *Keep It Simple, Stupid*. Togliere tutto ciò che non serve.

2. Sicurezza = conoscenza

- Nessun sistema sconosciuto può essere considerato sicuro
- Open source o closed source? Fiducia in una **comunità** che può leggere il codice, o in un'**azienda**
- Educare gli utenti è la prima forma di conoscenza

2. Il principio di Kerckhoffs

> *"Un sistema crittografico deve essere sicuro anche se tutto, tranne la chiave, è di pubblico dominio."*

il progetto (l'algoritmo)



pubblico: lo vedono tutti, anche Eve

la chiave



solo la chiave è segreta

il contrario: progetto segreto



«security by obscurity»

prima o poi trapela



Auguste Kerckhoffs (1835–1903)

3. Le componenti di un sistema



Un sistema sicuro oggi può non esserlo domani.

3. Humanware: le password più usate

Nel mondo

1. 123456
2. admin
3. 12345678
4. 123456789
5. 12345
6. password
7. Aa123456
8. 1234567890
9. Pass@123
10. admin123

In Italia

1. admin
2. password
3. 123456
4. Password
5. 12345678
6. 123456789
7. 12345
8. Napoli1926
9. 123stella
10. perlanera

NordPass, edizione 2025 (violazioni pubbliche tra settembre 2024 e settembre 2025, 44 Paesi).

3. Humanware: quanto pesano le password comuni

Su 6,5 milioni di credenziali trapelate, quanti usano una password "in classifica"?



M. Burnett, *10,000 Top Passwords*, 2011, valori ricalcolati dall'autore.

Proviamo: che password usi?

Prova una password

`juventus1897`

La password non lascia questo computer: il calcolo avviene nel browser.

Caratteri usati: ✓ **minuscole** (26) – maiuscole (26) ✓ **cifre** (10) – simboli (33)

Alfabeto e lunghezza	36 simboli possibili, 12 caratteri
Combinazioni	$36^{12} \approx 4{,}74 \times 10^{18}$
Forza bruta, caso peggiore	**15 anni** a 10^{10} tentativi al secondo
Password molto diffuse	✕ **parola della lista + numeri**

✕ **Parola comune + numeri: un attacco a dizionario la prova presto.**

Anche i dati personali aiutano: con nome, data di nascita e una password trapelata altrove si indovina in 100 tentativi la password di oltre il 73% degli utenti (Wang et al., ACM CCS 2016).

L’attacco a dizionario

Invece di provare tutte le combinazioni, si provano prima le password più diffuse.

Account **avv.bianchi** hash della password rubato dal database: `a941a4c4fd…f6c223`

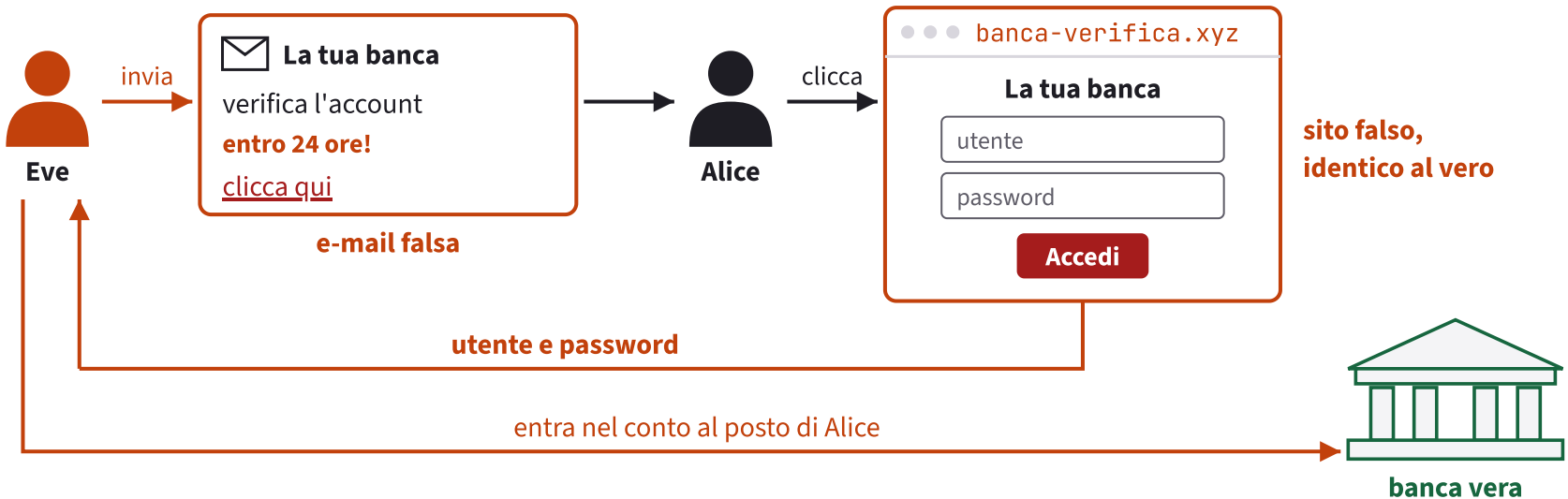
L'attaccante calcola l'hash di ogni parola del dizionario e lo confronta con l'hash rubato.

Tentativi: **20** su 64 parole del dizionario

```
tentativo 13: password1 → 0b14d5… ✗
tentativo 14: iloveyou → e4ad93… ✗
tentativo 15: 1q2w3e4r → 72ab99… ✗
tentativo 16: 000000 → 91b4d1… ✗
tentativo 17: qwerty123 → daaad6… ✗
tentativo 18: zaq12wsx → e3e93b… ✗
tentativo 19: dragon → a9c43b… ✗
tentativo 20: sunshine → a941a4… ✓
✓ trovata: sunshine (tentativo 20)
```

Difese (NIST, 2025): rifiutare le password comuni o già trapelate, limitare i tentativi, almeno 15 caratteri.

Phishing: la password te la consegnano

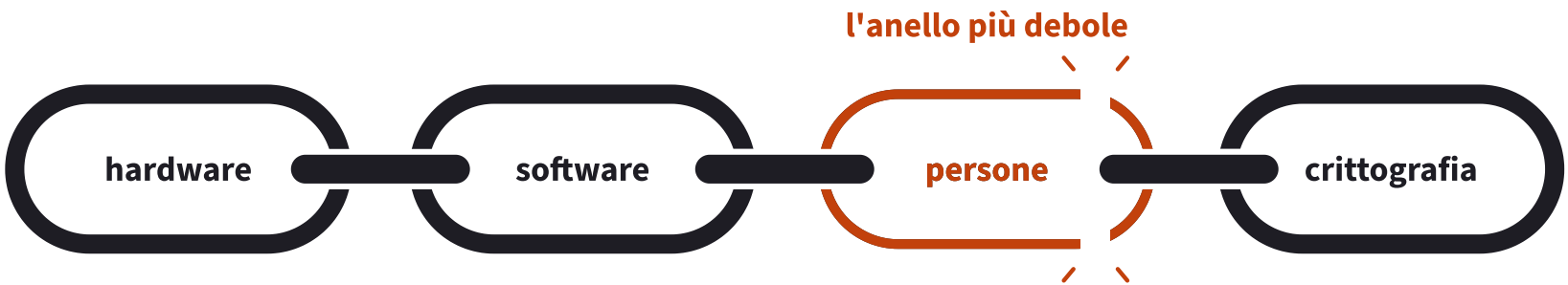


Febbraio 2025: una voce clonata del ministro Crosetto chiede soldi a imprenditori; uno versa circa 980.000 €, poi bloccati.

Video: una scena di phishing dalla serie *Mr. Robot*, youtube.com/watch?v=JMYEr4Bgey4

L’anello più debole

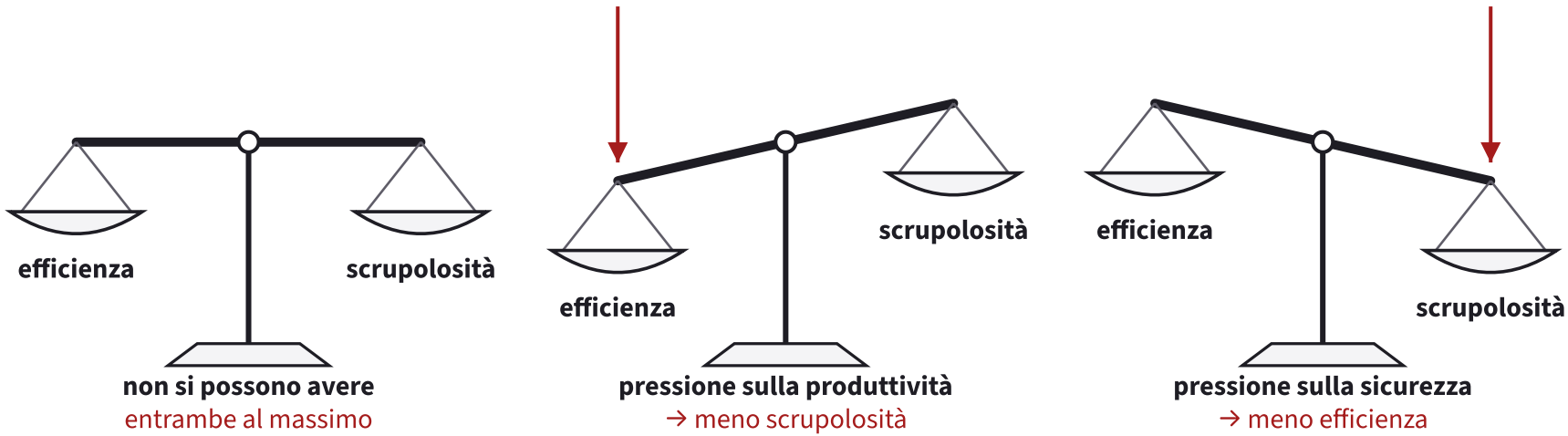
> "*Security is a chain; it's only as secure as the weakest link.*" B. Schneier, *Secrets and Lies*, 2000



la sicurezza è una catena: forte quanto l'anello più debole

Le persone sono davvero l'anello debole?

> *"La variabilità [dell'attività umana] è dunque una forza, anzi una necessità, più che un limite."* E. Hollnagel (trad.)



Le regole di sicurezza vanno pensate per persone vere, sotto pressione.

Verifica rapida

Un ransomware cifra i server di un ospedale: le cartelle cliniche non si aprono più. Quale proprietà è violata per prima?

- A Riservatezza
- B Integrità
- **C Disponibilità** ✓ giusta

> I dati ci sono ma non si possono usare: è la disponibilità. Se gli attaccanti li copiano e minacciano di pubblicarli (doppia estorsione) è violata anche la riservatezza.

Secondo il NIST, cosa rende più sicura una password usata da sola?

- A Cambiarla ogni 90 giorni
- **B Che sia lunga (almeno 15 caratteri) e non compaia tra le password comuni** ✓ giusta
- C Che contenga per forza un simbolo e una maiuscola

> Lunghezza e assenza dalle liste di password note contano più delle regole di composizione (NIST SP 800-63B-4).

II

Capo II

Crittografia

L’arte di scrivere in modo che solo il destinatario possa leggere

Crittografia

Le tecniche per cifrare un messaggio in modo che solo il legittimo destinatario possa leggerlo.

- **Efficiente** per chi è autorizzato: cifrare e decifrare è veloce
- **Difficile** per chi non lo è: troppo costoso o troppo lungo

> *Difficile non vuol dire impossibile: vuol dire che costa più di quanto valga il segreto.*

Crittografia applicata



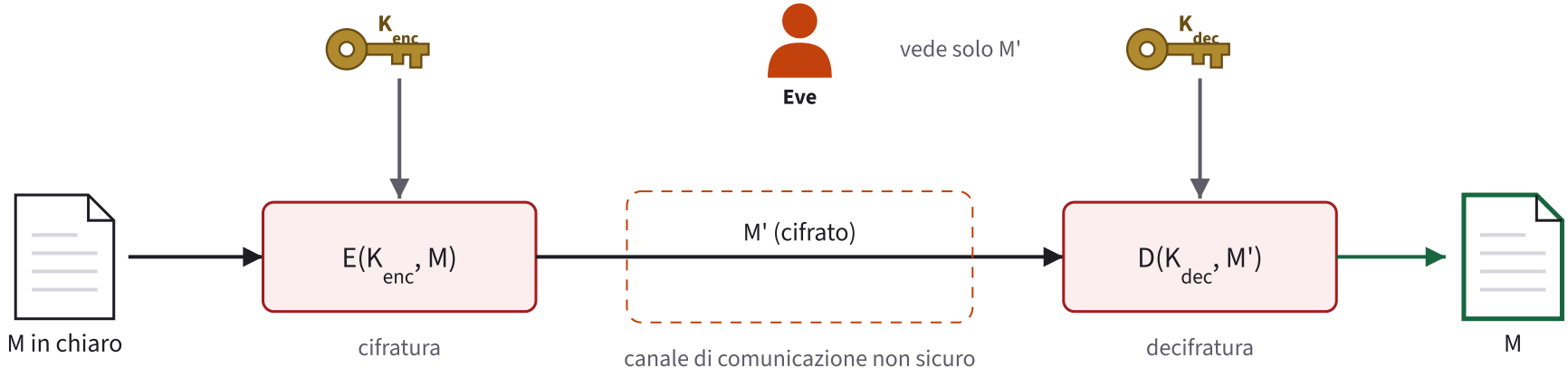
Il cilindro di Jefferson (anni 1790): l'ordine dei dischi è la chiave. Usato dall'esercito USA come M-94 dal 1922.



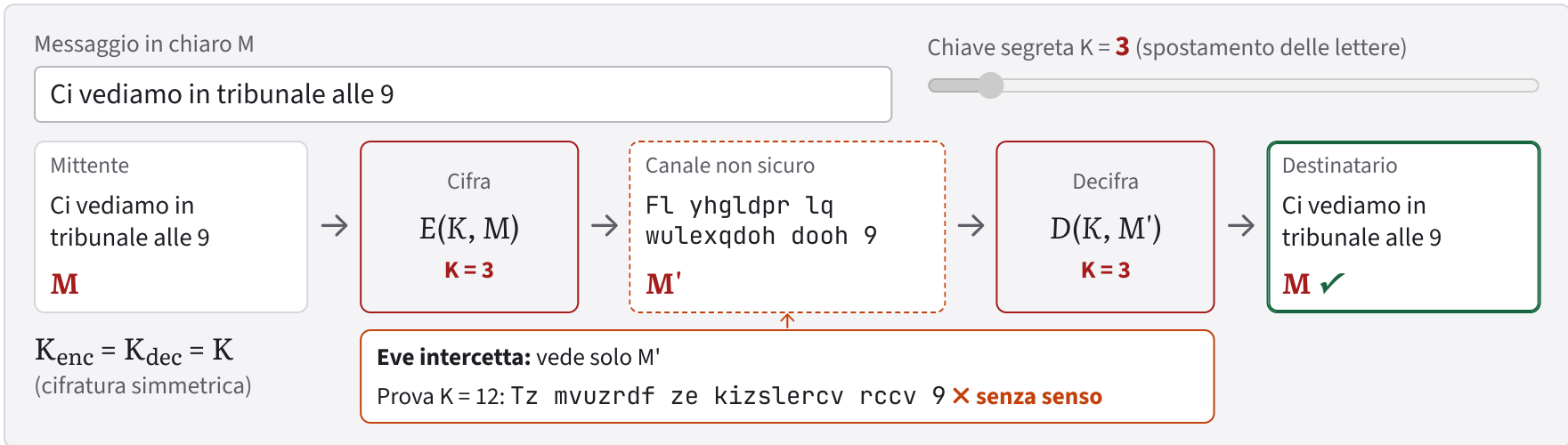
Enigma, forze armate tedesche, Seconda guerra mondiale. Forzata da polacchi e britannici. Museo della Scienza e della Tecnologia di Milano.

Foto: NSA, pubblico dominio; Alessandro Nassiri, CC BY-SA 4.0 (Wikimedia Commons).

Il principio di base



Proviamo: cifrare e decifrare

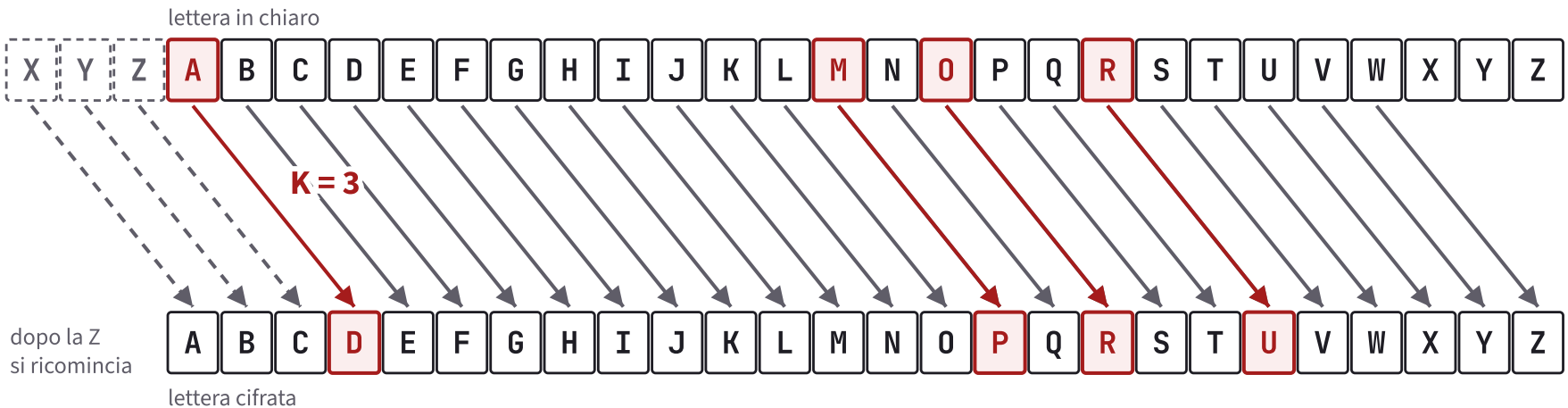


Notazione

- **M**: messaggio in chiaro (*plaintext*)
- **M'**: messaggio cifrato (*ciphertext*)
- **E(K_{enc}, M) = M'**: cifratura con la chiave K_{enc} (*encryption*)
- **D(K_{dec}, M') = M**: decifratura con la chiave K_{dec} (*decryption*)

Esempio: il cifrario di Cesare

Ogni lettera è sostituita da quella K posizioni più avanti. La chiave è il numero K.



ROMA → URPD

Proviamo: il cifrario di Cesare

Testo in chiaro

Alea iacta est

Chiave K = **3**: ogni lettera avanza di 3 posti

in chiaro	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
cifrato	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Testo cifrato **Dohd ldfwd hvw**

La regola d'oro

$$D(K_{dec}, E(K_{enc}, M)) = M$$

- Con la chiave giusta torna **esattamente** M
- Con una chiave sbagliata, solo **spazzatura**

Due famiglie di sistemi crittografici

A chiave segreta

crittografia simmetrica

$$K_{enc} = K_{dec}$$

Una chiave sola, con copie identiche: come la cassaforte.

A chiave pubblica

crittografia asimmetrica

$$K_{enc} \neq K_{dec}$$

Due chiavi diverse: come la cassetta delle lettere.

Verifica rapida

Il cifrario di Cesare con chiave K = 3 trasforma 'CASA' in…

A DBTB

B FDVD ✓ giusta

C ZXPX

D CASA3

> Ogni lettera avanza di 3 posizioni: C→F, A→D, S→V, A→D. Per decifrare si torna indietro di 3: stessa chiave, quindi è un sistema simmetrico.

Capo III

III Crittografia simmetrica

Una sola chiave, condivisa da mittente e destinatario

Cifratura simmetrica



Stessa chiave alle due estremità: è il "segreto condiviso" (*shared secret*).

Il cifrario di Cesare, attaccato

Le chiavi sono solo 25: basta provarle tutte. È l'attacco a forza bruta (*brute force*).

Testo cifrato intercettato (chiave sconosciuta)

`Dwwdffr doo'doed`

Proviamo tutte le 25 chiavi. Quale riga si legge?

K = 1 `Cvvceeq cnn'cndc`	K = 2 `Buubddp bmm'bmcb`	K = 3 **`Attacco all'alba`**	K = 4 `Zsszbbn zkk'zkaz`
K = 5 `Yrryaam yjj'yjzy`	K = 6 `Xqqxzzl xii'xiyx`	K = 7 `Wppwyyk whh'whxw`	K = 8 `Voovxxj vgg'vgwv`
K = 9 `Unnuwwi uff'ufvu`	K = 10 `Tmmtvvh tee'teut`	K = 11 `Sllsuug sdd'sdts`	K = 12 `Rkkrttf rcc'rcsr`
K = 13 `Qjjqsse qbb'qbrq`	K = 14 `Piiprrd paa'paqp`	K = 15 `Ohhoqqc ozz'ozpo`	K = 16 `Nggnppb nyy'nyon`
K = 17 `Mffmooa mxx'mxnm`	K = 18 `Leelnnz lww'lwml`	K = 19 `Kddkmmy kvv'kvlk`	K = 20 `Jccjllx juu'jukj`
K = 21 `Ibbikkw itt'itji`	K = 22 `Haahjjv hss'hsih`	K = 23 `Gzzgiiu grr'grhg`	K = 24 `Fyyfhht fqq'fqgf`
K = 25 `Exxeggs epp'epfe`			

✓ Trovata: K = 3, al massimo 25 tentativi.

Una chiave che cambia

Cambiare la chiave a ogni messaggio, o addirittura a ogni lettera: lo stesso testo dà cifrati diversi.

Messaggio

`Hello`

Stesso messaggio, chiave diversa ogni volta: **testi cifrati diversi**. Chi intercetta non capisce nemmeno che il messaggio si ripete.

Cifratura n. 1
spostamento +3
`Hello` → **`Khoor`**

Cifratura n. 2
spostamento −1
`Hello` → **`Gdkkn`**

Cifratura n. 3
spostamento +2
`Hello` → **`Jgnnq`**

Cifratura n. 4
spostamento +6
`Hello` → **`Nkrru`**

Vigenère: la chiave cambia a ogni lettera

Parola chiave `LEX` — L = 11, E = 4, X = 23 (A = 0 … Z = 25)

Messaggio	H	e	l	l	o
Lettera della chiave	L	E	X	L	E
Spostamento	11	4	23	11	4
Risultato	**S**	**i**	**i**	**w**	**s**

La stessa lettera del messaggio può diventare lettere diverse: le frequenze delle lettere non si riconoscono più.

DES: Data Encryption Standard

- Standard USA dal **1977**, progettato da IBM: chiave di **56 bit**
- $2^{56} \approx 7{,}2 \times 10^{16}$ chiavi: nel **1998** la macchina *Deep Crack* ne trova una in 56 ore
- Oggi hardware dedicato le prova tutte in circa un giorno
- Il **Triplo DES** non è più ammesso dal NIST per cifrare (dal 2024)

I computer accelerano: le chiavi devono allungarsi.

AES: Advanced Encryption Standard

- Standard dal **2001** (NIST FIPS 197), sostituisce DES
- Chiavi da **128, 192 o 256 bit**; blocchi da 128 bit
- $2^{128} \approx 3{,}4 \times 10^{38}$ chiavi: provarle tutte è impraticabile
- Regge anche ai computer quantistici: il NIST non prevede di sostituirlo

Protegge ogni giorno il Wi-Fi, i dischi dei portatili, WhatsApp, HTTPS.

Quanto tempo serve per provare tutte le chiavi?

Password | Chiavi

età dell'universo ≈ $1{,}38 \times 10^{10}$ anni

Alfabeto: solo minuscole (26)

Lunghezza: **8** caratteri

Chi attacca: tentativi al secondo: un computer: 1 miliardo al secondo

$26^8 \approx 2{,}09 \times 10^{11}$ combinazioni → nel caso peggiore **3,5 minuti**

Password	Combinazioni	un computer (10^9/s)	un grande impianto dedicato (10^{13}/s)
8 caratteri, minuscole	$26^8 \approx 2{,}09 \times 10^{11}$	3,5 minuti	istantaneo
10 caratteri, minuscole	$26^{10} \approx 1{,}41 \times 10^{14}$	1,6 giorni	14 secondi
12 caratteri, minuscole	$26^{12} \approx 9{,}54 \times 10^{16}$	3 anni	2,7 ore
12 caratteri, lettere e cifre	$62^{12} \approx 3{,}23 \times 10^{21}$	102.234 anni	10 anni
15 caratteri, lettere e cifre	$62^{15} \approx 7{,}69 \times 10^{26}$	$2{,}44 \times 10^{10}$ anni	$2{,}44 \times 10^{6}$ anni

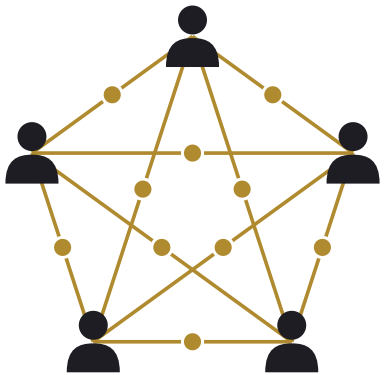
Crittografia simmetrica: pro e contro

A favore: è molto **efficiente**, anche su grandi quantità di dati.

- **Distribuzione delle chiavi**: come ci si accorda in segreto sulla chiave?
- **Scalabilità**: serve una chiave per ogni coppia di persone
- **Niente non ripudio**: la chiave ce l'hanno entrambi

Quante chiavi servono?

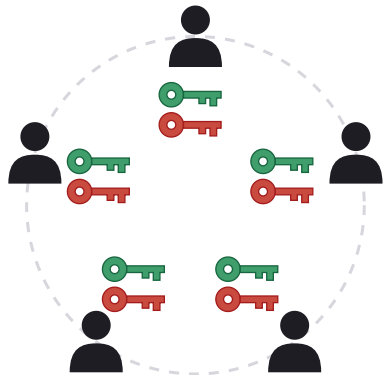
simmetrica



10 chiavi segrete

una per ogni coppia di persone

asimmetrica



5 coppie di chiavi

una coppia per persona

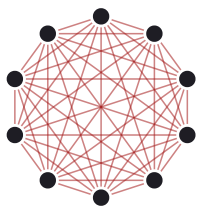
persone	simmetrica	asimmetrica
5	10	5
10	45	10
100	4.950	100
1.000	499.500	1.000

simmetrica: $n(n-1)/2$ chiavi

cresce con il quadrato di n

Proviamo: quante chiavi per n persone?

Persone che vogliono scriversi in privato, a due a due: **10**

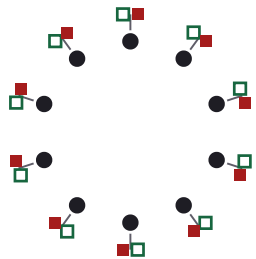


Cifratura simmetrica

45

chiavi segrete condivise

una per ogni coppia: n(n − 1)/2



Cifratura asimmetrica

10

coppie di chiavi

una per persona: 10 pubbliche + 10 private

pubblica privata

Il numero di chiavi simmetriche cresce con il quadrato di n (non esponenzialmente): il doppio delle persone, circa il quadruplo delle chiavi.

Con n = 1.000: **499.500** chiavi segrete contro **1.000** coppie di chiavi.

IV

Capo IV

Crittografia asimmetrica

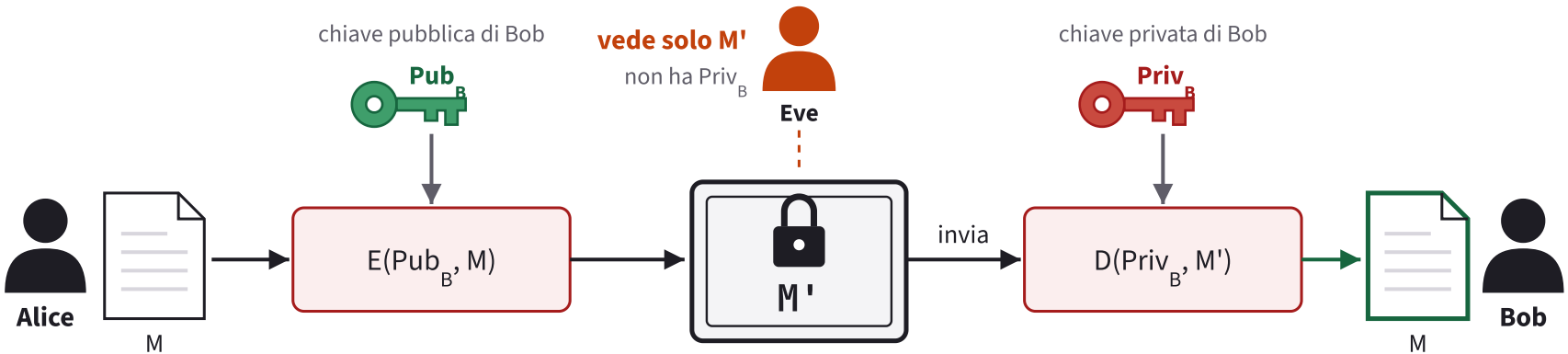
Una chiave per tutti, una chiave solo per me

Crittografia a chiave pubblica



Ciò che una chiave chiude, solo l'altra lo apre. E dalla pubblica non si ricava la privata.

Alice scrive a Bob, in segreto



Proviamo: Alice cifra, Bob decifra

Alice scrive a Bob

Il cliente accetta la transazione a 50.000 €

Alice cifra con la chiave pubblica di Bob

Messaggio cifrato, lo vede chiunque sulla rete (256 byte):

```
b8bbce3c2f38bd75cf3be29ac6fa
ae5f3d88bedb…0d813464
```

Bob, il destinatario

pubblica `cd1a b3c5 b11f d611` può conoscerla chiunque

privata la conosce solo Bob

Bob decifra con la sua chiave privata

✓ **«Il cliente accetta la transazione a 50.000 €»**

Eve intercetta il messaggio

privata di Eve è un'altra chiave, non quella di Bob

Eve prova con la sua chiave privata

✕ **impossibile decifrare**

Cosa si ottiene con le due chiavi

Cifro con la pubblica del destinatario

- **Riservatezza**: solo lui può leggere

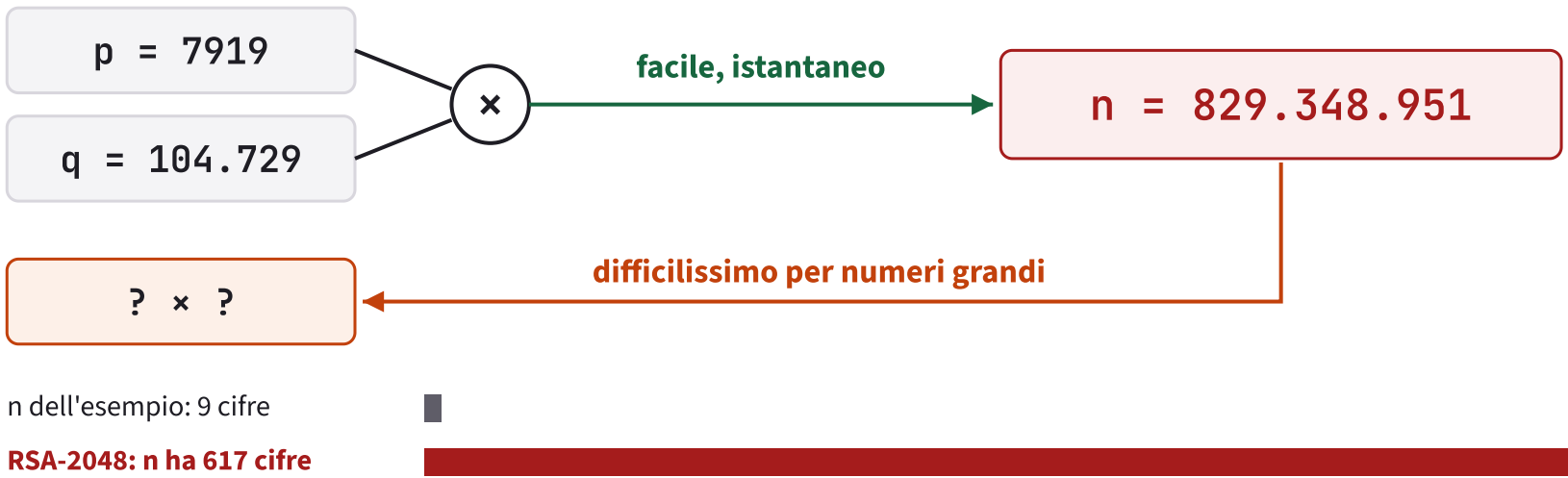
Firmo con la mia privata

- **Autenticità**: si sa chi ha scritto
- **Integrità**: ogni modifica si vede
- **Non ripudio**: non posso negare

La colonna di destra è la firma digitale: la vediamo nel Capo VII.

RSA (Rivest–Shamir–Adleman, 1977)

La sicurezza si basa, in parte, su un fatto pratico: moltiplicare è facile, fattorizzare è difficile.



Settembre 2026: nuovi record di fattorizzazione; RSA-2048 resta fuori portata.

Proviamo: moltiplicare e fattorizzare

Moltiplicare è facile

p 7919 × q 104.729 = **n = 829.348.951**

9 cifre, calcolato all'istante

Fattorizzare è difficile

Chi conosce solo n prova a dividerlo per 2, 3, 5, 7, 9, … finché trova un divisore.

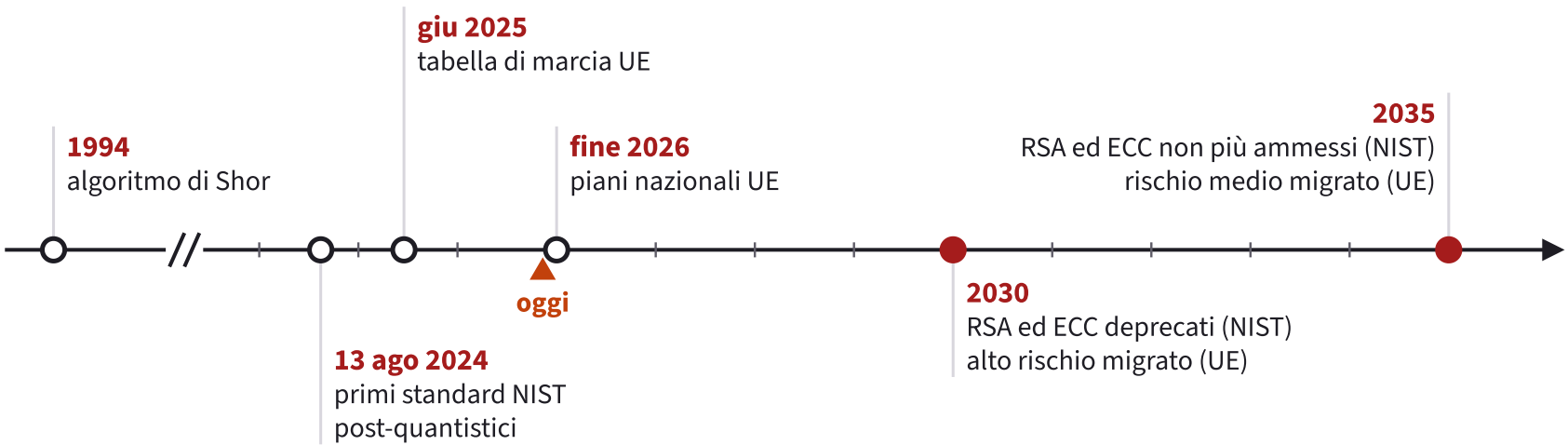
tentativi **3960**

✓ n = 7919 × 104.729, trovato dopo 3960 tentativi

Una chiave RSA-2048 ha un modulo di **617 cifre decimali**; qui n ne ha 9.

E il computer quantistico?

Con l'algoritmo di Shor, un computer quantistico abbastanza grande romperebbe RSA e le curve ellittiche. Oggi non esiste, ma ci si prepara.

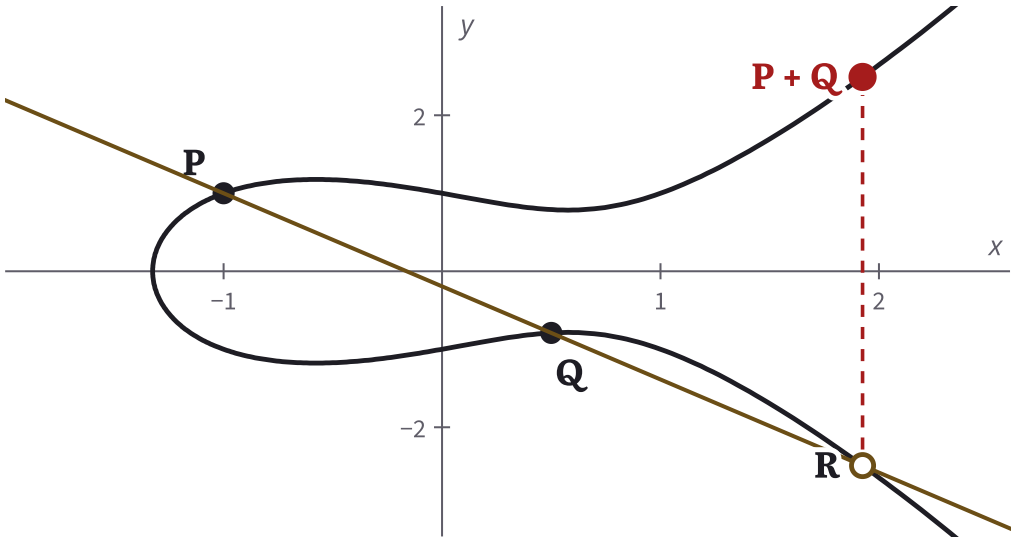


"Raccogli ora, decifra dopo": ciò che oggi viene intercettato cifrato potrà essere letto tra 10–15 anni.

ECDSA: firme su curve ellittiche

- Stessa sicurezza di RSA con **chiavi molto più corte**: 256 bit ≈ RSA da 3072 bit
- Usata da **Bitcoin** ed **Ethereum**: chi ha la chiave privata ha i fondi
- Variante moderna: **Ed25519**, approvata dal NIST nel 2023
- Anche le curve cadrebbero con un computer quantistico: arriva ML-DSA (FIPS 204)

La geometria delle curve ellittiche



$$y^2 = x^3 - x + 1$$

sui numeri reali, per poterla disegnare

P = (−1; 1)
Q = (0,5; $-\sqrt{0{,}625}$) ≈ (0,5; −0,79)

la retta per P e Q taglia la curva
in un terzo punto: R ≈ (1,92; −2,49)

si ribalta R sull'asse x:

P + Q ≈ (1,92; 2,49)

Verifica rapida

Alice vuole mandare a Bob un documento riservato. Con quale chiave lo cifra?

- A La chiave privata di Alice
- **B** La chiave pubblica di Bob ✓ **giusta**
- C La chiave privata di Bob

Con la chiave pubblica di Bob: solo la chiave privata di Bob, che nessun altro possiede, potrà decifrarlo.

Perché la crittografia asimmetrica risolve la distribuzione delle chiavi?

- A Perché le chiavi sono più lunghe
- **B** Perché la chiave pubblica si può diffondere senza rischi ✓ **giusta**
- C Perché non usa chiavi

La chiave pubblica può viaggiare in chiaro: non serve un canale sicuro per accordarsi prima su un segreto comune.

V

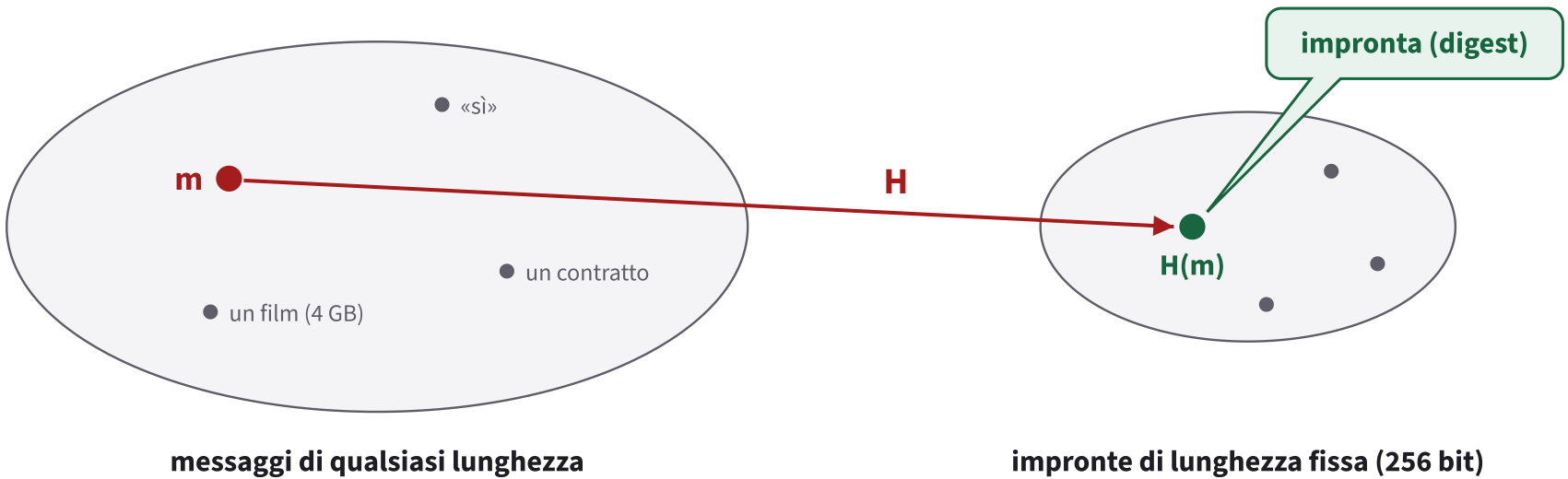
Capo V

Funzioni hash

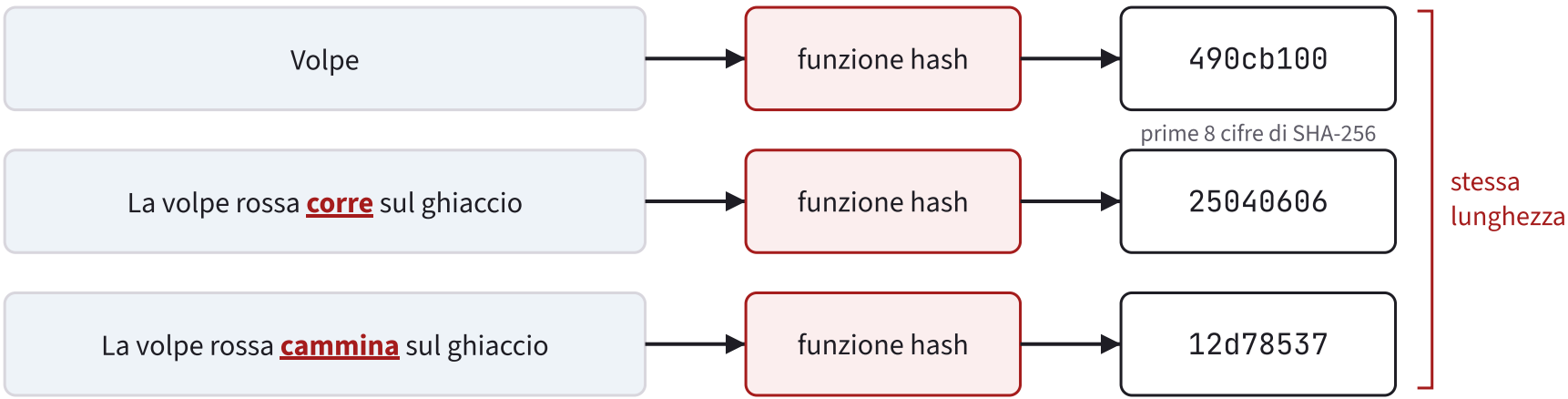
L’impronta digitale di un documento

Funzione hash

Trasforma dati di qualsiasi dimensione in un valore di dimensione fissa: l'impronta (*digest*).



Funzioni hash: esempi



lunghezza fissa, nessuna somiglianza

basta cambiare una parola e l'uscita cambia del tutto

Proviamo: calcolare un’impronta

Testo A

La volpe

`01a1d88cb53073a8315394c983e96499935452407 24a4f356bbff4f56c3109a2`

Testo B

La volpe rossa corre sul ghiaccio

`2504060688fae70ced883c5b1749e6597a13236c67198e304a1daf8633fbb409`

Bit diversi: **120** su 256 (47%)

Funzioni hash crittografiche

Funzioni hash progettate per resistere a chi cerca di ingannarle. Si usano per:

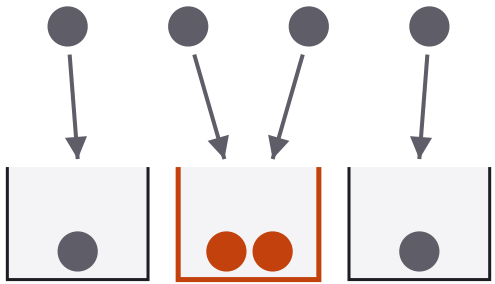
- verificare l'**integrità** di messaggi e file
- la **firma digitale** (si firma l'impronta)
- verificare le **password** (si conserva l'impronta)
- il **proof-of-work** delle blockchain

Le proprietà di una funzione hash crittografica

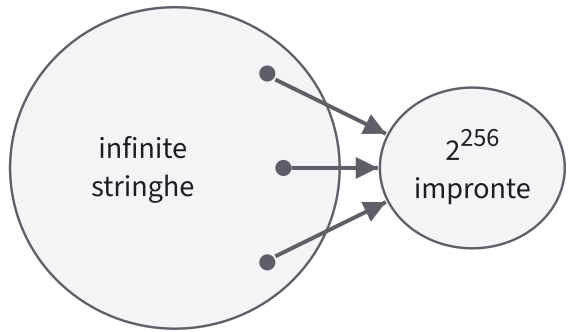
1. **Resistenza alle collisioni**: impossibile in pratica trovare due input con la stessa impronta
2. **Occultamento** (*hiding*): dall'impronta non si risale all'input
3. **Adatta ai puzzle** (*puzzle-friendly*): nessuna scorciatoia, solo tentativi

1) Resistenza alle collisioni

principio dei cassetti

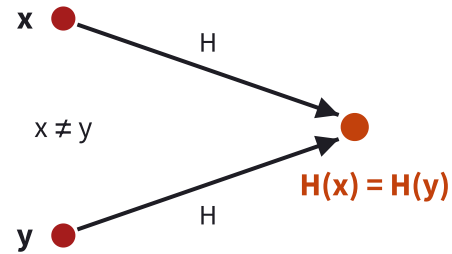


4 oggetti in 3 cassetti: uno ne contiene 2



le collisioni esistono!

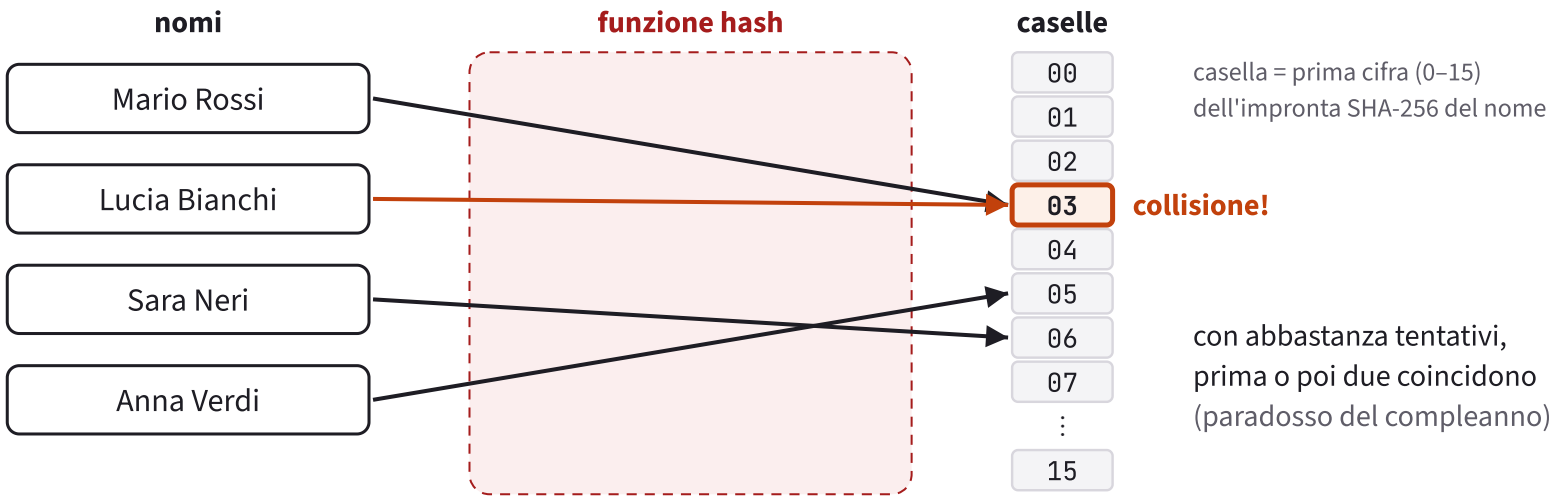
una collisione



due testi diversi, stessa impronta

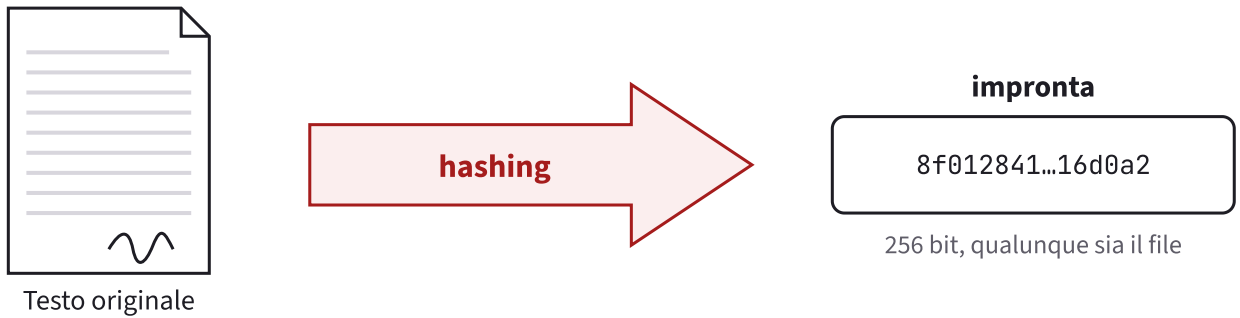
le collisioni ~~non esistono~~ **esistono, ma nessuno sa trovarle**

1) Come si trova una collisione?



- Con 2^{130} input a caso, la probabilità di una collisione supera il 99,8%
- Ma anche a 10^{18} impronte al secondo servirebbero migliaia di volte l'età dell'universo

1) Applicazione: identificare un file

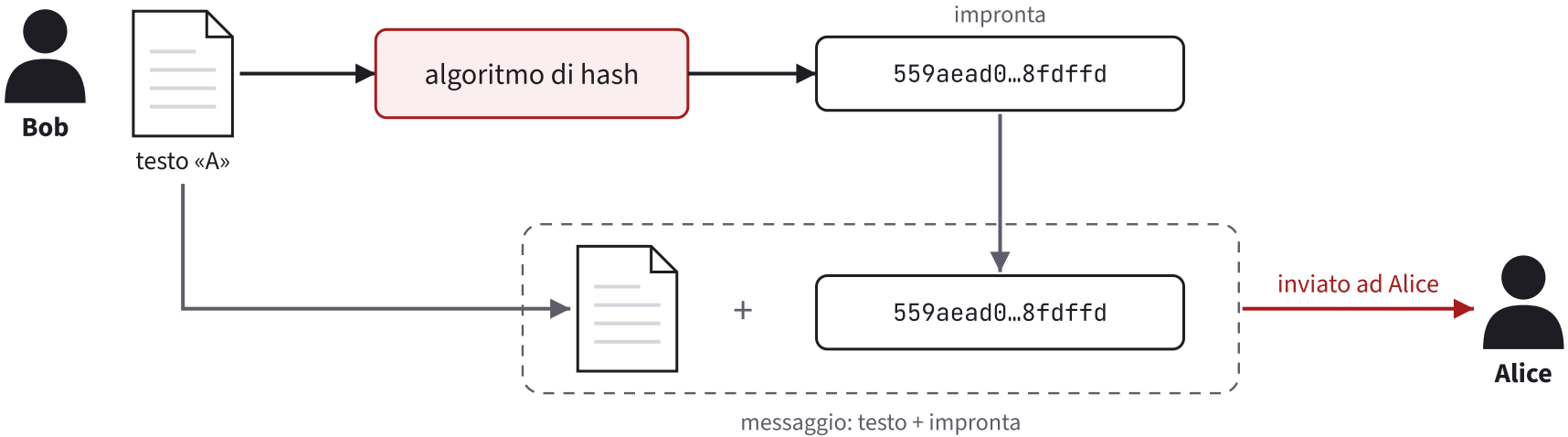


per riconoscere un file basta la sua impronta

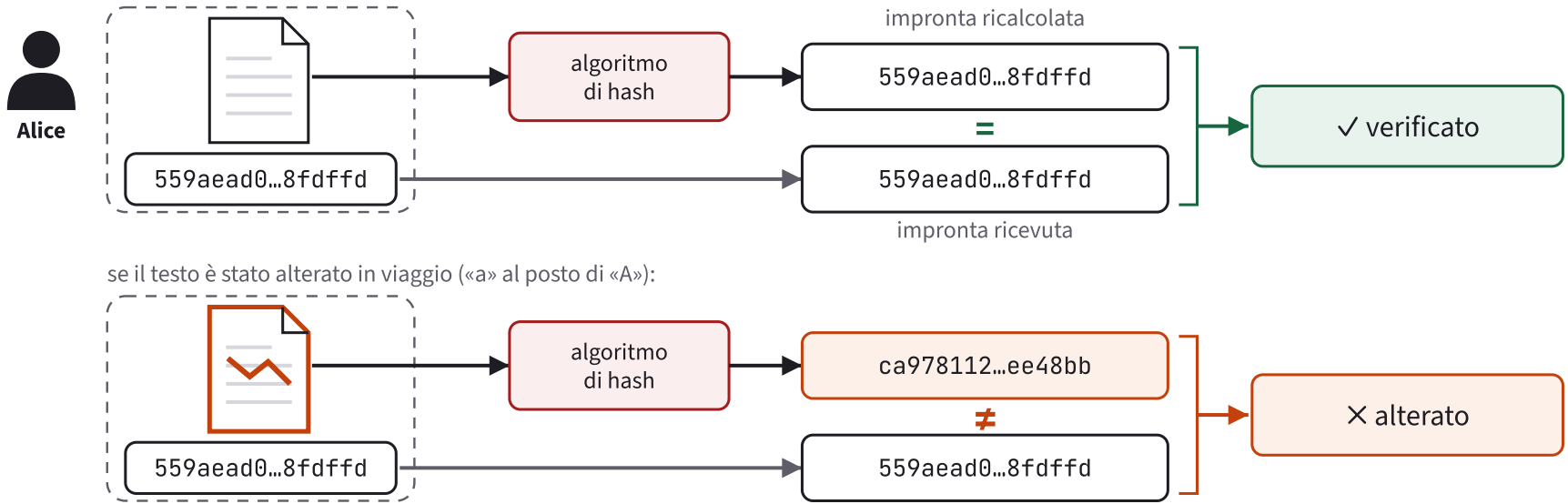
stessa impronta ⇒ stesso file (se l'hash resiste alle collisioni)

Se H(x) = H(y), possiamo assumere che x = y: per riconoscere un file basta la sua impronta.

1) Applicazione: integrità dei dati



1) Integrità dei dati: la verifica



L'hash **rivela** l'alterazione, non la **impedisce**.

Proviamo: un intruso modifica il contratto

Bob invia il documento e la sua impronta

Contratto di locazione: canone mensile 800 €

Impronta inviata `260451102fba4fa5dea1135b75e168c8798c37aede1e55e9d3e3d16c16fd69b4`

Alice riceve la copia intatta — **✓ integro: le impronte coincidono**

Contratto di locazione: canone mensile 800 €

Impronta ricalcolata `260451102fba4fa5dea1135b75e168c8798c37aede1e55e9d3e3d16c16fd69b4`

Alice riceve la copia modificata dall'intruso — **✗ alterato: le impronte sono diverse**

Contratto di locazione: canone mensile 900 €

Impronta ricalcolata `7c377fd091a2971f076f2b86ff2c57351097471ac8a63714961f6442ea06790b`

Se l'intruso può cambiare anche l'impronta, il solo hash non basta: serve la firma digitale (Capo VII).

2) Occultamento (*hiding*)

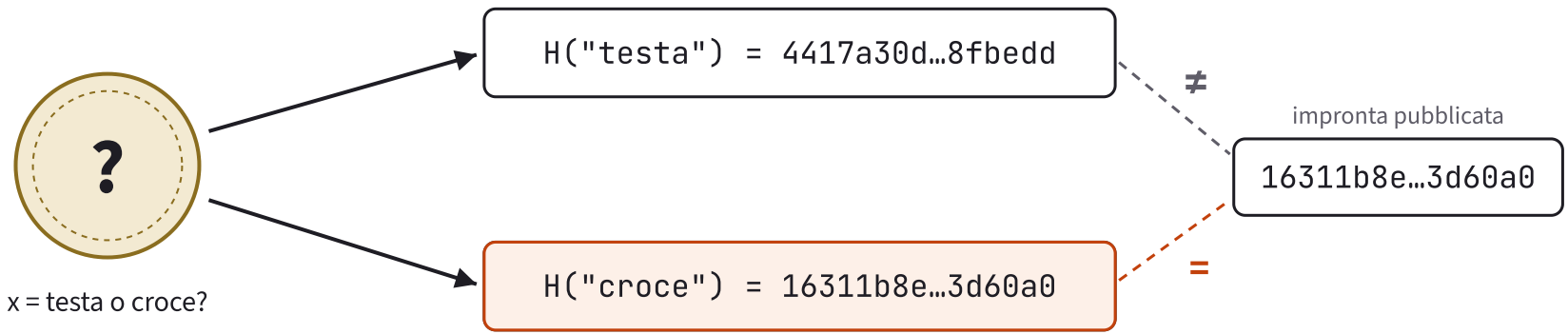


stessa lunghezza: sempre 256 bit (64 cifre; qui ne mostriamo 32)

dall'impronta non si risale al testo

(purché i testi possibili siano moltissimi)

2) Quando l'occultamento fallisce



scoperto: era «croce». Con 2 possibilità basta provarle

Proviamo: testa o croce

Ho scelto **testa** o **croce** e tengo la scelta nascosta: pubblico solo l'impronta SHA-256.

Senza segreto

Impronta pubblicata H(scelta) **`16311b8e4f9c…3d60a0`**

H(«testa»)	`4417a30dc8c6…8fbedd`	≠ diversa
H(«croce»)	`16311b8e4f9c…3d60a0`	= **uguale**

✕ Scelta scoperta, era «croce»: con solo due possibilità, l'impronta non nasconde nulla.

Con un segreto casuale (nonce)

Impronta pubblicata H(nonce ‖ scelta)

`ec90d6c48ca7…c400c9`

nonce segreto di 128 bit

H(«testa»)	`4417a30dc8c6…8fbedd`	≠ diversa
H(«croce»)	`16311b8e4f9c…3d60a0`	≠ diversa

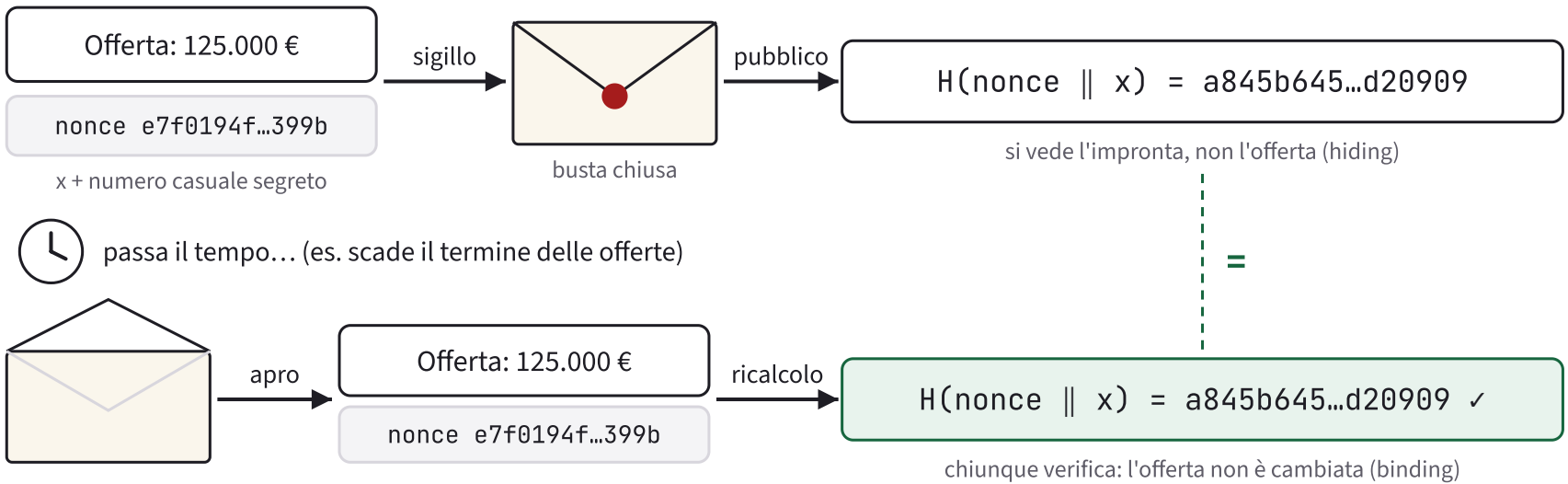
✓ Nessuna corrisponde: senza il nonce l'impronta non rivela la scelta.

2) L'occultamento funziona solo se...

- i valori possibili di x sono **moltissimi**
- nessuno è **più probabile** degli altri
- nessuno è **più interessante** degli altri

Rimedio: aggiungere un segreto casuale (*nonce*) e pubblicare H(nonce ‖ x).

2) Applicazione: impegno (*commitment*)



Proviamo: l'offerta in busta chiusa

Chi offre, in privato

Offerta

Offerta: 125.000 €

Segreto casuale (nonce, 128 bit): `7c1e9a4b3f…7d3a52`: lo conosce solo chi offre

Tutti gli altri

Busta chiusa: si pubblica solo l'impronta H(nonce ‖ offerta)

`845797782898d6f91ec55e306b4c2b17a59b8bf0047e325551ec4f10b81be845`

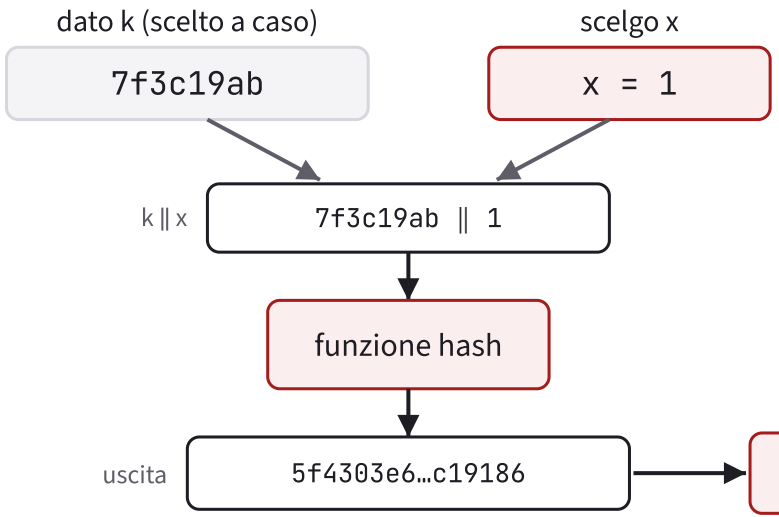
Si rivelano l'offerta «**Offerta: 125.000 €**» e il nonce `7c1e9a4b3f…7d3a52`

Impronta ricalcolata

`845797782898d6f91ec55e306b4c2b17a59b8bf0047e325551ec4f10b81be845`

✓ **L'impronta corrisponde: è proprio l'offerta sigillata.**

3) Adatta ai puzzle (*puzzle-friendly*)

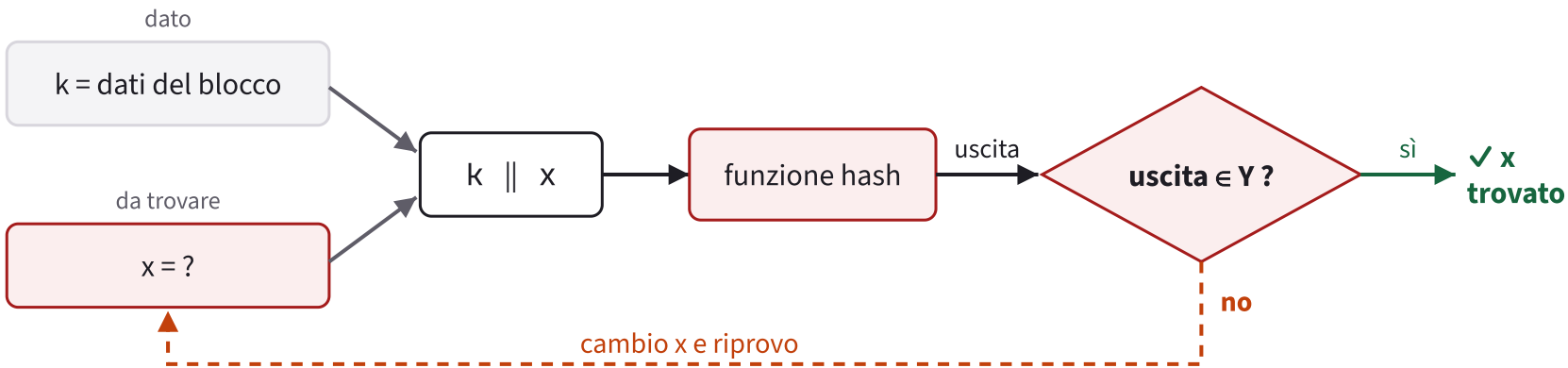


tentativi

x = 1	→	5f4303e6…c19186	✕
x = 2	→	1af49d98…b26b8c	✕
x = 3	→	344f0e51…188d41	✕
⋮			
x = 85349	→	0000843d…af853a	✓

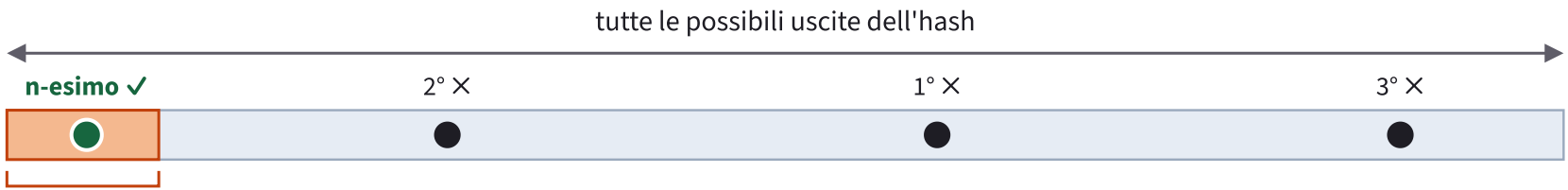
nessuna scorciatoia:
si può solo provare x = 1, 2, 3…

3) Applicazione: il puzzle di ricerca



Y = insieme bersaglio (per esempio: le uscite che iniziano con 0000)

3) Come lanciare una moneta



come un lancio di moneta: ogni tentativo cade in un punto a caso

con un Y più piccolo:



più Y è piccolo, più il puzzle è difficile

servono in media molti più tentativi

Proviamo: risolvere un puzzle

Dati del blocco k

Blocco 1: Alice paga Bob 10

Difficoltà d

4 zeri

Trova un numero x tale che l'impronta H(k + x) inizi con **0000**. Si può solo tirare a indovinare: in media servono 16^4 = **65.536** tentativi.

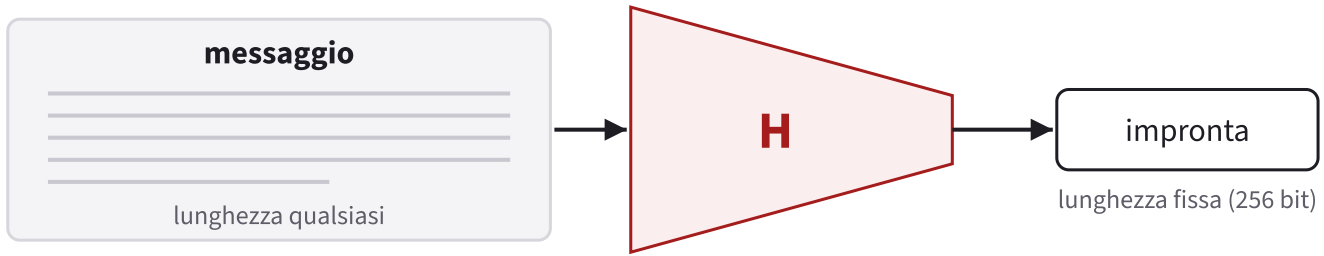
tentativi **288.160**

H(k + 288159)

0000**67296e9abe8e46316df031dbb74b4c7e6ef0a68a767ea98d5f454387b5b5**

✓ **Trovato: x = 288159, dopo 288.160 tentativi.** Verificarlo invece richiede un solo calcolo.

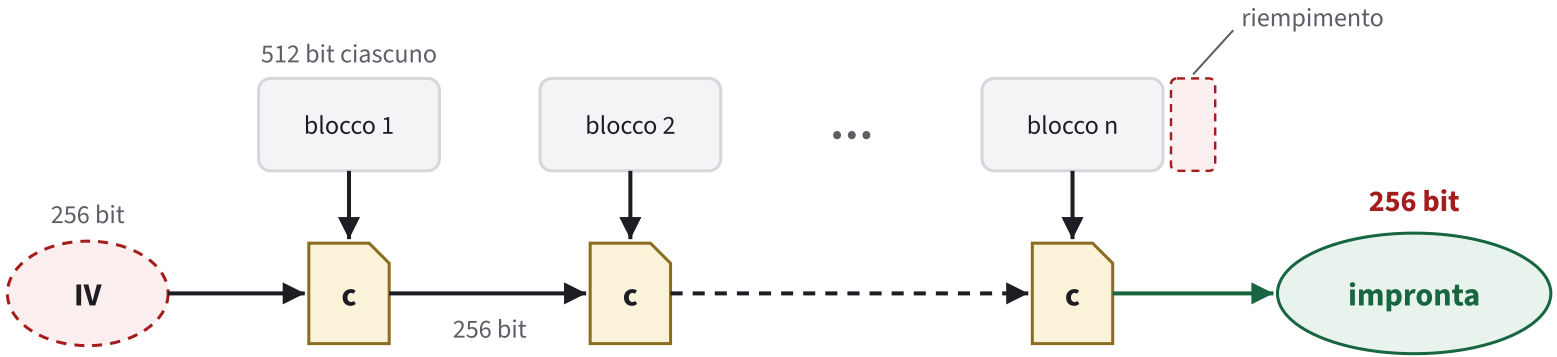
Funzioni hash: schemi esistenti



- **MD5** (128 bit): **rotta**, una collisione costa una frazione di secondo
- **SHA-1** (160 bit): **rotta**, prima collisione reale nel 2017 (SHAttered)
- **SHA-2** (SHA-256/384/512): **raccomandata**
- **SHA-3** (Keccak, 2015): **raccomandata**

La slide del 2024 diceva che per SHA-1 c'erano solo "attacchi teorici": non è più vero dal 2017.

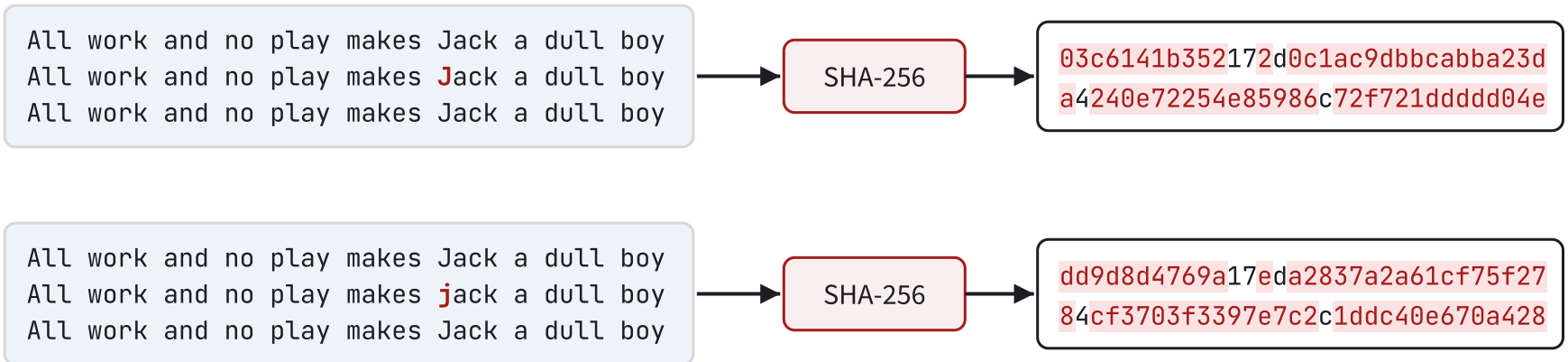
SHA-256: come è fatta dentro



IV = vettore di inizializzazione (un valore fisso, sempre lo stesso)

c = funzione di compressione

Esempio: SHA-256



una lettera diversa: cambiano 59 cifre su 64 (135 bit su 256)

le cifre uguali sono poche e capitano per caso (effetto valanga)

Esempio: SHA-256 dal terminale

Terminale

```
$ echo "ciao" | sha256sum
6f0378f21a495f5c13247317d158e9d51da45a5bf68fc2f366e450deafdc8302 -
$ echo "miao" | sha256sum
549d63fbe569a4dcb200dec0d65786794c931d4a976e8644f1035695117b20e9 -
$ echo "ciao mare" | sha256sum
079542b606786b70eb79bed391e8672331bdf6f09fa0c5e1349cdd0faa768687 -
$ echo "ciao care" | sha256sum
a945dd5628c2864e297162adf2b400d8d04f3088582aae03ccea5a054c5e5a26 -
$ printf "ciao" | sha256sum
b133a0c0e9bee3be20163d2ad31d6248db292aa6dcb1ee087a2aa50e0fc75ae2 -
```

echo aggiunge un a capo: per questo l'impronta è diversa da quella di «ciao» senza a capo

Proviamo: SHA-256 dal vivo

Testo A

All work and no play makes Jack a dull boy
All work and no play makes Jack a dull boy
All work and no play makes Jack a dull boy

`f80f3a6b5ae6d5f33fce632cfb867e98843cbd6f7976feebe7e4cf29b4a37659`

Testo B

All work and no play makes Jack a dull boy
All work and no play makes jack a dull boy
All work and no play makes Jack a dull boy

`65e0375fdaf60711368914051a53c8dceb74af8c6c7bc251200dccf4eb51e264`

Bit diversi: **126** su 256 (49%)

☐ aggiungi "a capo" finale (come fa `echo`)

Verifica rapida

Un perito calcola l'impronta di un hard disk sequestrato e, mesi dopo, quella della copia: coincidono. Cosa può concludere?

- **A** Che la copia è identica all'originale — ✓ **giusta**
- **B** Che nessuno ha mai letto i dati
- **C** Niente: le collisioni esistono

> Le collisioni esistono in teoria, ma trovarne una è impraticabile: impronte uguali significano, in pratica, dati identici. L'impronta non dice invece nulla su chi ha letto i dati.

Perché non basta pubblicare H(«sì») per tenere segreto un voto sì/no?

- **A** Perché SHA-256 è lento
- **B** Perché chiunque può calcolare H(«sì») e H(«no») e confrontare — ✓ **giusta**
- **C** Perché l'impronta è troppo lunga

> Con due soli input possibili l'occultamento fallisce: serve un segreto casuale (nonce) da aggiungere al voto.

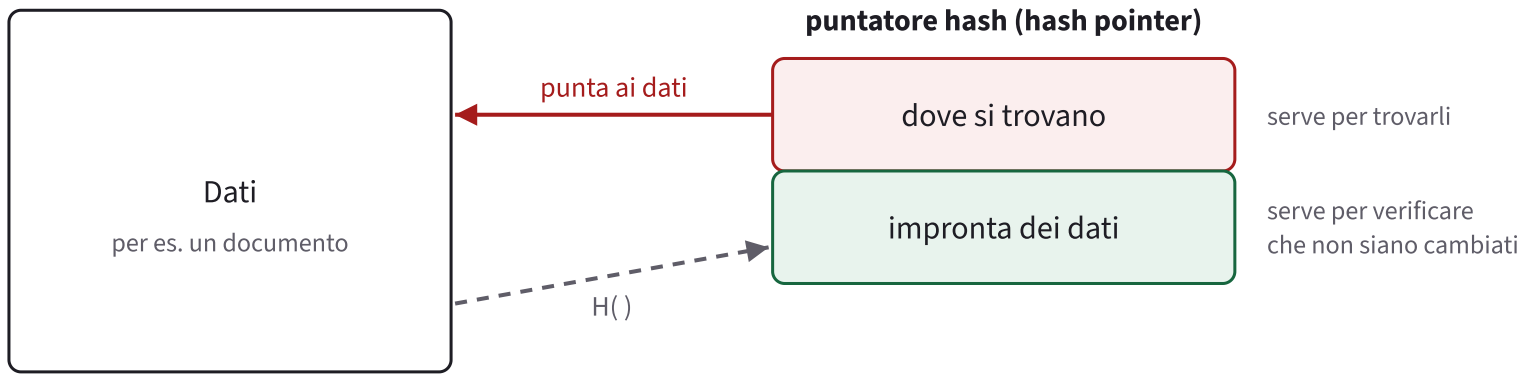
Capo VI

VI

Puntatori hash e strutture dati

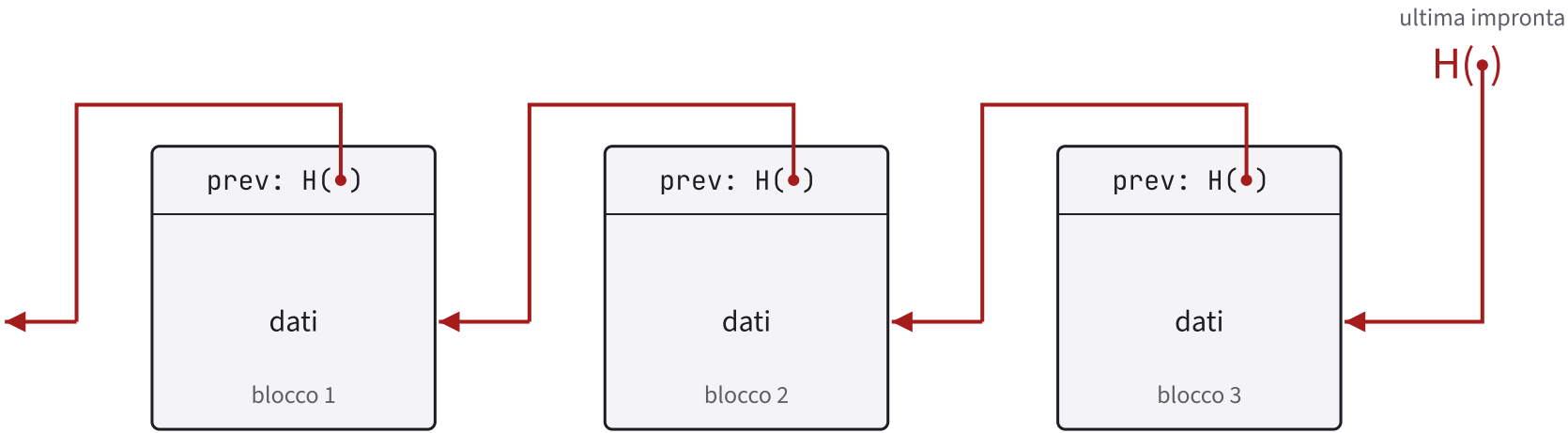
Come incatenare i dati in modo che ogni manomissione si veda

Puntatore hash (*hash pointer*)



Dice **dove** si trova un dato e **com'era**: se qualcuno lo cambia, l'impronta non torna.

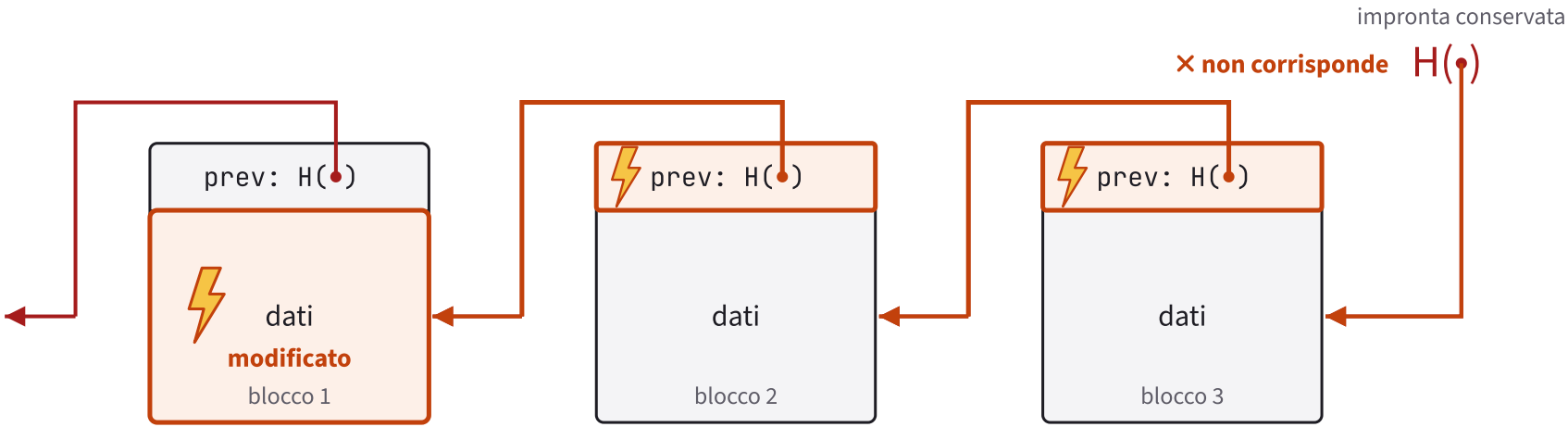
L’idea chiave: catene di puntatori hash



basta conservare l'ultima impronta

È la struttura che Bitcoin, nel 2008, ha reso famosa: la **blockchain**.

Rilevare la manomissione



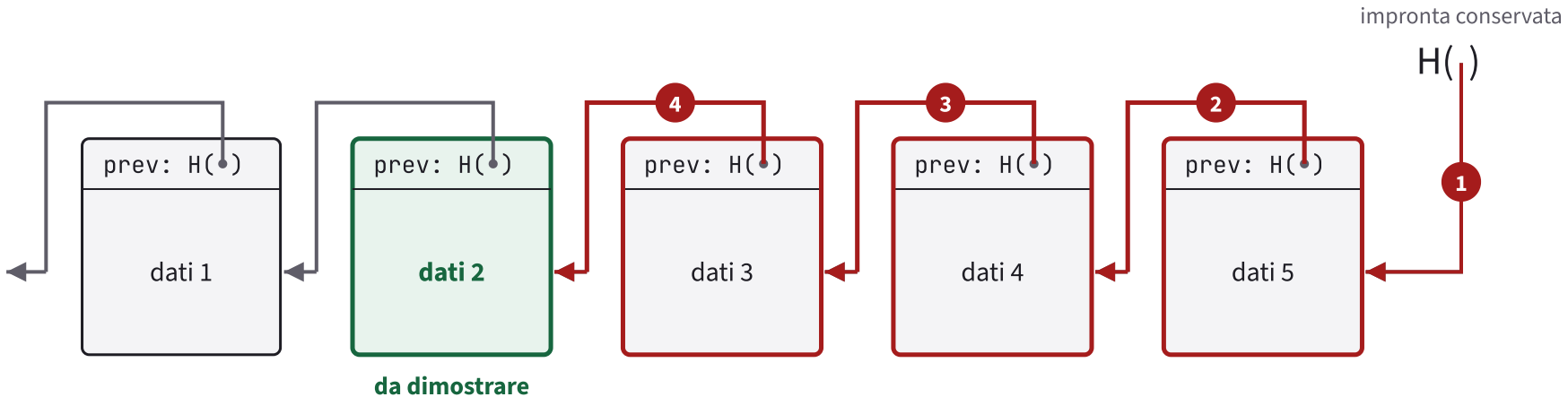
E se l'attaccante riscrive anche i prev? Cambia l'impronta di ogni blocco successivo…

si arriva sempre all'impronta conservata: la modifica si vede

Proviamo: modificare un verbale



Dimostrare che un elemento è nella catena

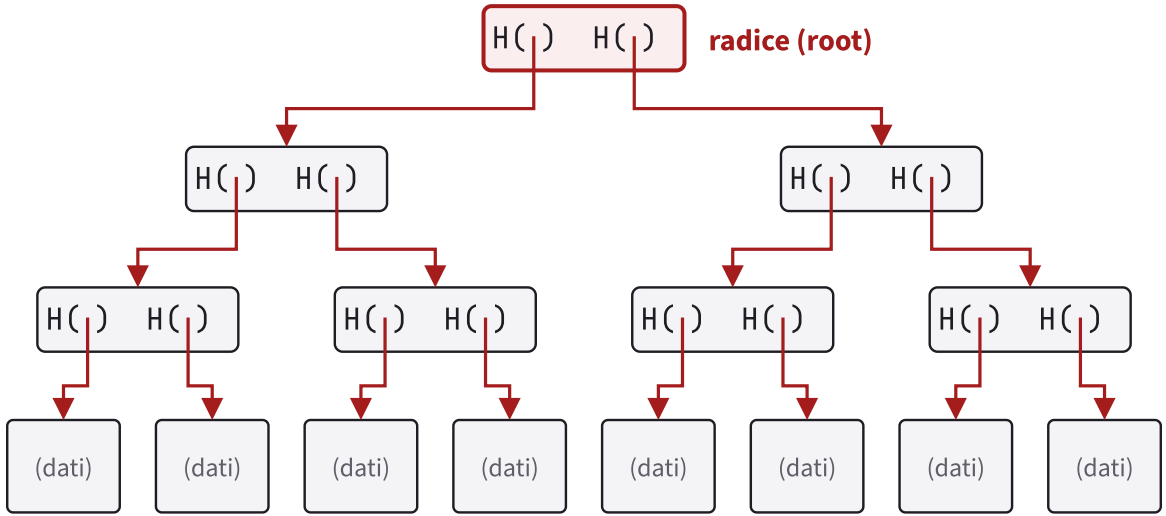


costo proporzionale a n: O(n)

con n blocchi, nel caso peggiore servono n passi

Con un milione di elementi, nel caso peggiore, un milione di passaggi. Si può fare di meglio?

L’albero di Merkle



quanti livelli per n dati?

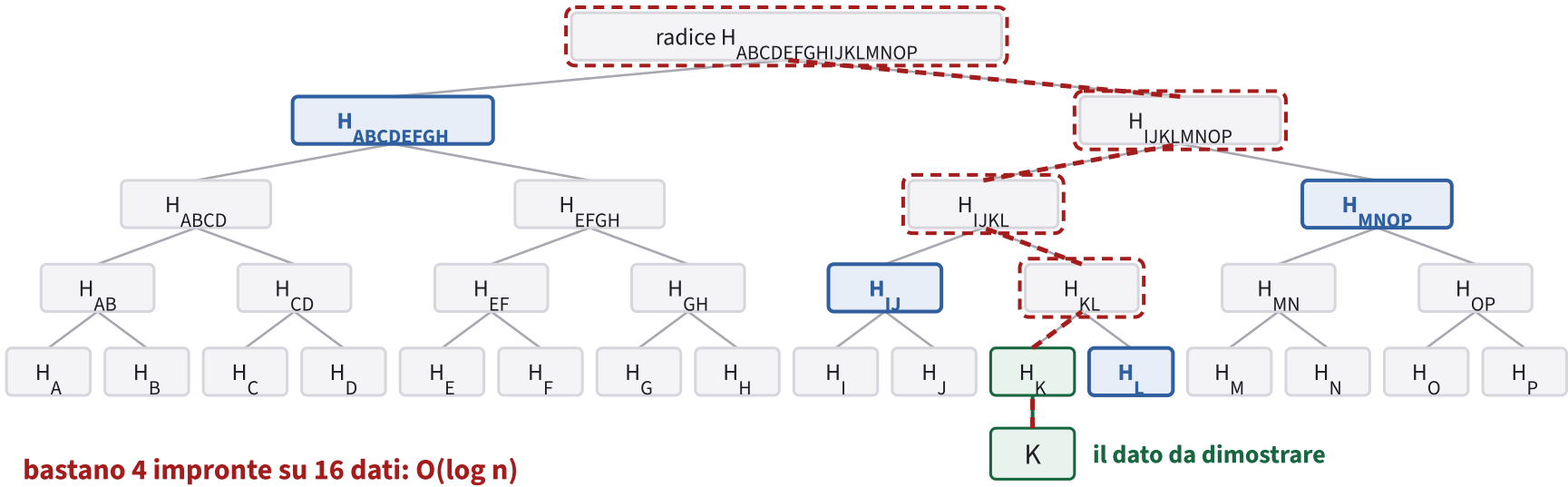
$\log_2 n$ (8 dati → 3 livelli)

impronta delle due sottostanti

transazioni (dati)

Ralph Merkle, brevetto USA 4.309.569 (domanda del 1979, concesso nel 1982).

Dimostrare l'appartenenza in un albero di Merkle

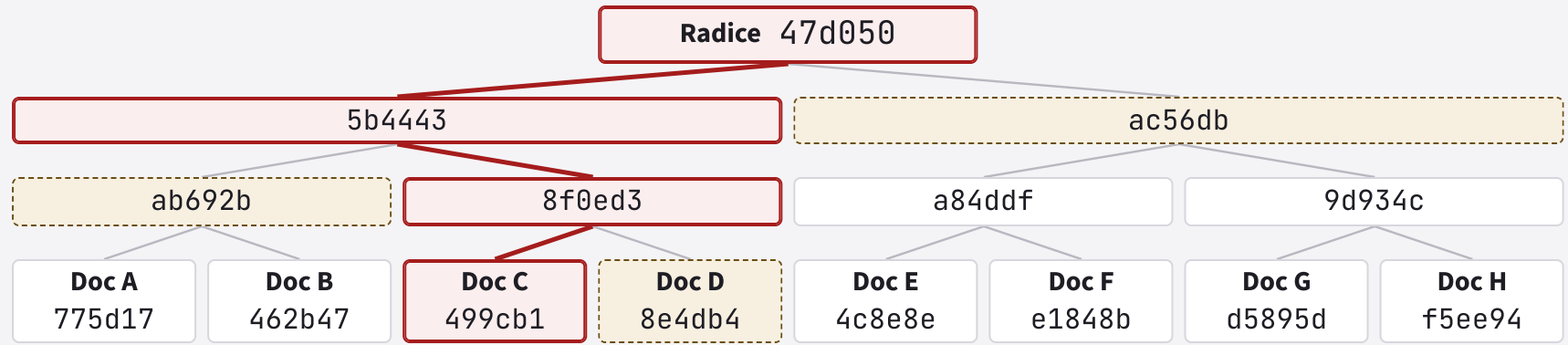


bastano 4 impronte su 16 dati: O(log n)

tratteggiato: ciò che si ricalcola, da K fino alla radice

in blu: le impronte fornite con la prova

Proviamo: la prova di Merkle



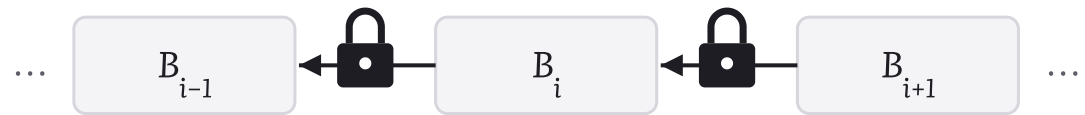
Per dimostrare che 'Doc C' è incluso bastano **3 hash** ($\log_2 8$), non tutti gli 8 documenti.

Modifica il documento selezionato: Doc C

percorso fino alla radice — hash della prova (forniti con il documento)

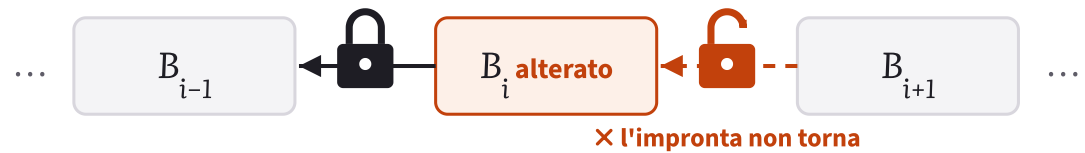
Hash camaleontico (*chameleon hashing*)

catena originale



senza la chiave privata

qualcuno altera B_i



con la chiave privata

la "botola" (trapdoor)



modificabile solo da chi ha la chiave privata:
utile per il diritto alla cancellazione, ma è un punto di fiducia

Verifica rapida

In una catena di puntatori hash qualcuno modifica il blocco 2 di 4. Cosa succede?

A Nulla di visibile

B Il collegamento del blocco 3 non torna più ✓ giusta

C Si cancellano tutti i blocchi

Il blocco 3 conserva l'impronta del blocco 2 originale: non corrisponde più. Per nascondere la modifica bisognerebbe ricalcolare tutta la catena fino all'impronta conservata altrove.

Un albero di Merkle contiene 1.024 documenti. Quante impronte servono per dimostrare che uno è incluso?

A 1.024

B 512

C 10 ✓ giusta

$\log_2 1.024 = 10$: una impronta sorella per ogni livello dell'albero.

Capo VII

VII Non ripudio e firma digitale

Dalla matematica delle chiavi al valore giuridico della firma

Non ripudio nella sicurezza digitale

La prova dell'integrità e dell'origine dei dati, verificabile anche da un terzo.



Alice non può negare di averlo firmato

solo chi ha S_{Alice} può produrre una firma che P_{Alice} accetta

L’integrità

- L’impronta permette di **accorgersi** di qualunque alterazione
- La verifica la fa il **destinatario**, ricalcolando l’impronta

Correzione rispetto al 2024: l'hash non rende "improbabile" l'alterazione, la rende **rilevabile**.

La verifica dell'origine

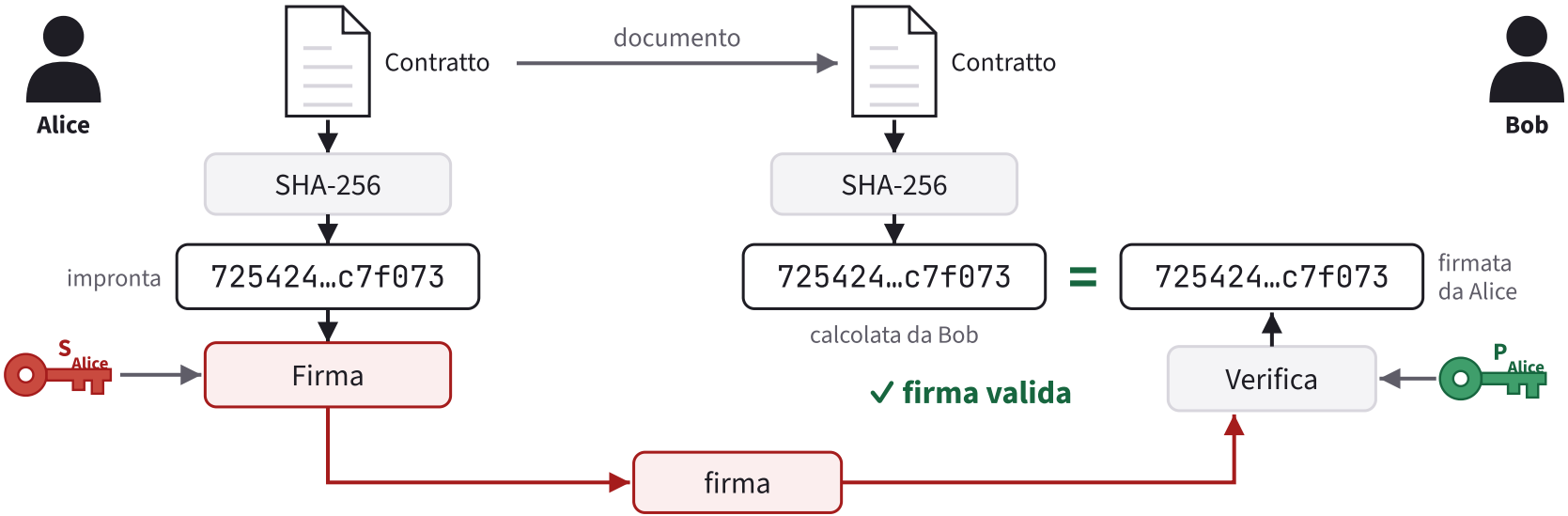
- **Firma digitale**: prova che il documento è stato firmato con una certa chiave privata
- **Certificato digitale**: dice a chi appartiene quella chiave
- I certificati li emette un'autorità, dentro un'**infrastruttura a chiave pubblica** (PKI)

Firma digitale

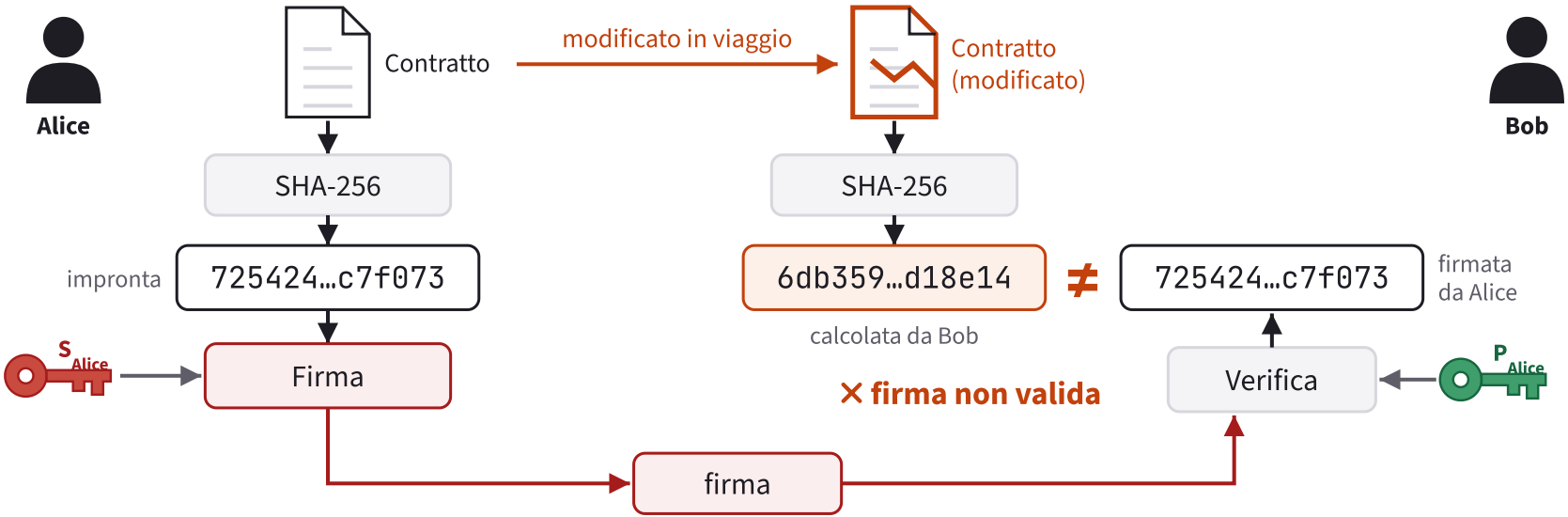
- Il firmatario usa la sua **chiave privata**; chiunque verifica con la **chiave pubblica**
- **Integrità**: il documento non è cambiato dopo la firma
- **Autenticazione**: la firma viene dal titolare della chiave

Non è l'immagine di una firma scansionata: è un calcolo sull'impronta, diverso per ogni documento.

Alice firma un documento per Bob



Alice firma un documento (alterato) per Bob



La firma resta legata a **quel** documento: non si stacca, non sopravvive a una modifica.

Proviamo: firmare e verificare

Alice firma

Documento di Alice

Il sottoscritto Mario Rossi accetta la proposta del 3 novembre 2026.

1. Alice calcola l'impronta (SHA-256)

Impronta: `7a936f6a7f6d2ccd9b6567fbe0d77cb86abdf1a3b179a3a7eb6324644363819b`

2. Alice firma con la sua chiave privata

Firma: `7e68ba01fcdba3…ed60a5`

→

Bob verifica

Documento ricevuto da Bob

Il sottoscritto Mario Rossi accetta la proposta del 3 novembre 2026.

Chiave pubblica di Alice: `04b9d179…b56e`

Modifica una parola in transito

3. Bob verifica con la chiave pubblica di Alice

✓ firma valida: documento integro, firmato da Alice

Se in transito «accetta» diventa «rifiuta»: **✕ firma non valida: documento alterato o firmatario diverso**

Certificato digitale



Alice

Certificato digitale

Nome	Alice
Indirizzo	via Galliera 3
Email	alice@studio-legale-esempio.it
Chiave pubblica	

… ma chi garantisce che dica il vero?

lega una chiave pubblica a un'identità

Infrastruttura a chiave pubblica (PKI) e X.509

- **X.509**: il formato più comune dei certificati (RFC 5280)
- **PKI**: le autorità e le regole che garantiscono a chi appartiene una chiave
- Il lucchetto di **HTTPS**: il browser verifica un certificato X.509 a ogni sito

Il certificato firmato dall'autorità di certificazione

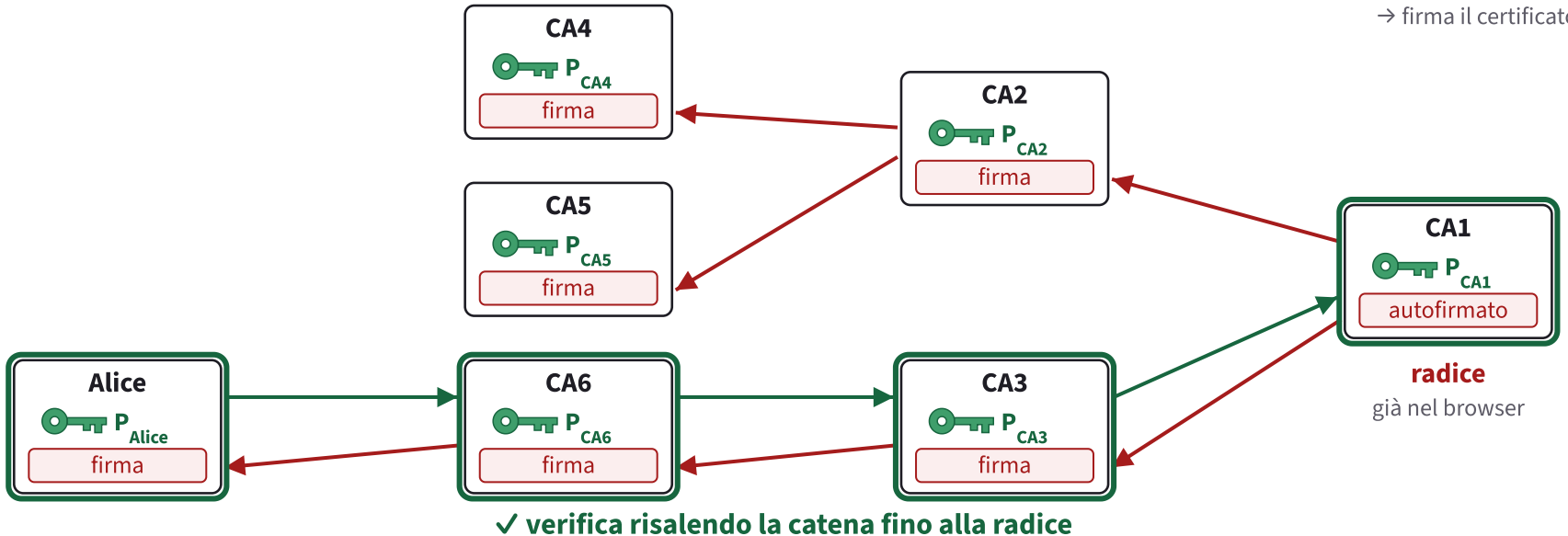


Dentro un certificato X.509



Nel browser: clic sul lucchetto accanto all'indirizzo per vedere questi campi su un sito vero.

La catena di fiducia X.509



Proviamo: verificare una catena

Certificato del sito
Lo invia il sito quando ti colleghi

Soggetto
www.studio-legale-esempio.it
Emittente
Intermedia Esempio CA
Valido dal / al
01/10/2026 – 18/04/2027
Chiave pubblica `04e8ce…fbd5`
Firma dell'emittente `e4abd5…8ced`

← firma

Autorità intermedia
Firmata dall'autorità radice

Soggetto
Intermedia Esempio CA
Emittente
Radice Esempio CA
Valido dal / al
13/03/2024 – 12/03/2027
Chiave pubblica `043edc…062e`
Firma dell'emittente `80e589…6e1b`

← firma

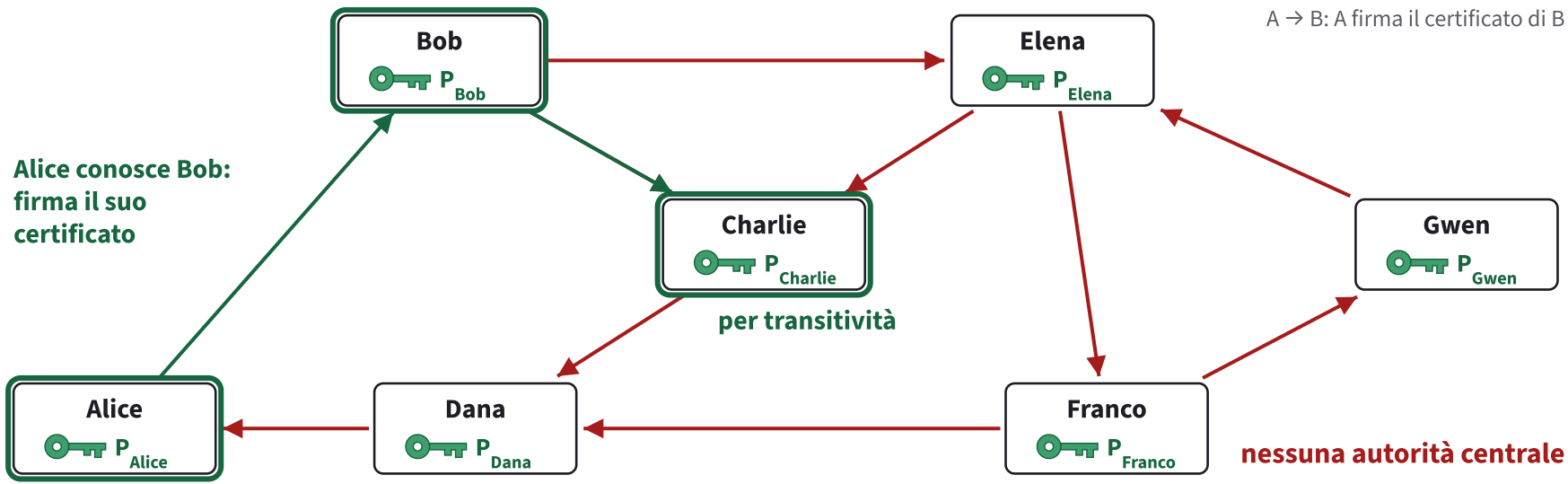
Autorità radice
Già presente nel tuo browser/sistema operativo

Soggetto
Radice Esempio CA
Emittente
Radice Esempio CA
Valido dal / al
04/06/2015 – 04/06/2035
Chiave pubblica `04eac9…cec2`
Firma dell'emittente `863874…6563`

✓ **catena valida: il browser mostra il lucchetto**

- ✓ il certificato di «www.studio-legale-esempio.it» è valido oggi (fino al 18/04/2027)
- ✓ la firma di «www.studio-legale-esempio.it» è verificata con la chiave pubblica di «Intermedia Esempio CA»
- ✓ la firma di «Intermedia Esempio CA» è verificata con la chiave pubblica di «Radice Esempio CA»
- ✓ la radice è fidata perché è già nel tuo computer

Un'alternativa: la rete di fiducia (PGP)



Provate voi: firmare e verificare

Lo strumento ufficiale della Commissione europea per le firme eIDAS (DSS).

- "Validate a signature": carica un .p7m o un PDF firmato e leggi il rapporto
- "Sign a document": firma in XAdES, CAdES, PAdES o JAdES (serve l'app NexU)

ec.europa.eu/digital-building-blocks/DSS/webapp-demo

eIDAS: il regolamento europeo

- **Reg. (UE) 910/2014**: identificazione elettronica e servizi fiduciari
- Regole comuni per firme, sigilli e validazioni temporali in tutta l'UE

> *"Le firme elettroniche qualificate basate su un certificato qualificato rilasciato in uno Stato membro [...] sono riconosciute [...] in tutti gli altri Stati membri."* art. 24 bis, par. 1

eIDAS 2 e il portafoglio digitale

- Ogni Stato deve offrire un **portafoglio digitale** (EUDI Wallet) entro il **24 dicembre 2026**
- Dovrà permettere di **firmare con firma qualificata, gratis**
- In Italia: **IT-Wallet**, nell'app IO

eIDAS riconosce tre tipi di firma elettronica

1 Firma elettronica (semplice)
dati elettronici usati per firmare
valore probatorio ●○○

2 Avanzata (FEA)
+ legata al firmatario
+ sotto il suo controllo
+ modifiche rilevabili
valore probatorio ●●○

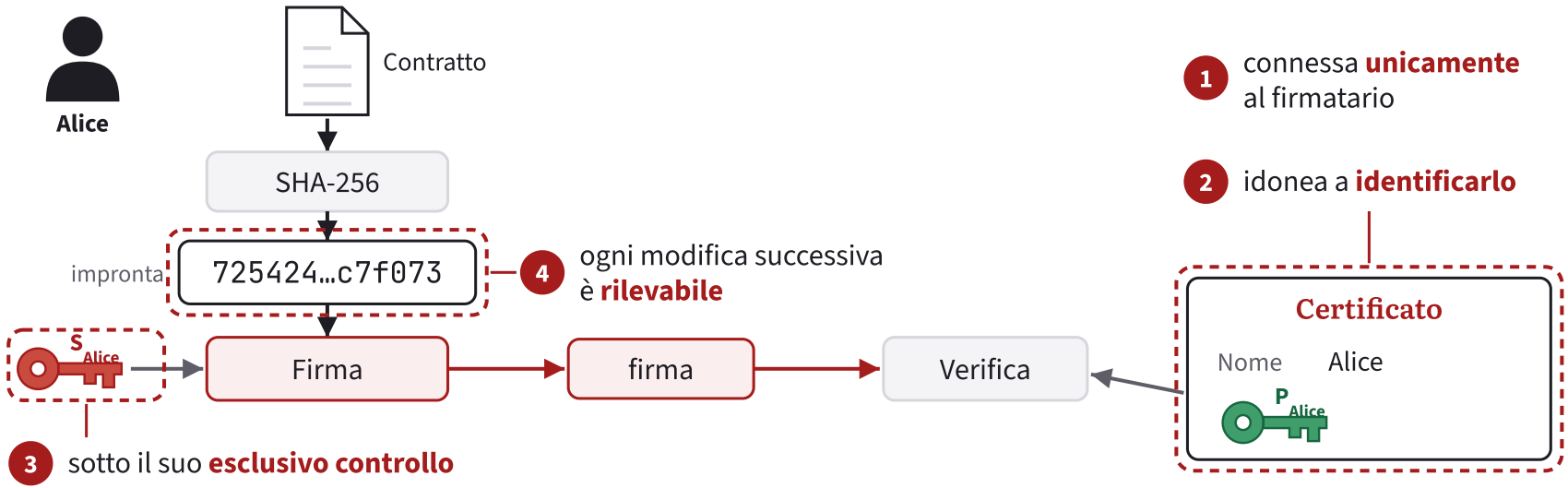
3 Qualificata (FEQ)
+ certificato qualificato
+ dispositivo qualificato
= firma autografa
valore probatorio ●●●

1. Firma elettronica (semplice)

> *"A una firma elettronica non possono essere negati gli effetti giuridici [...] per il solo motivo della sua forma elettronica."* art. 25, par. 1, eIDAS

Esempi: il nome in fondo a una email; il clic su "Accetto"; un PIN.

2. Firma elettronica avanzata (FEA)



Esempi: XAdES, CAdES, PAdES senza certificato qualificato; la firma grafometrica su tablet in banca.

FEA: i formati

Un servizio pubblico online deve accettare le firme avanzate almeno in questi formati.

XAdES	**CAdES**	**PAdES**
XML	il file .p7m	PDF

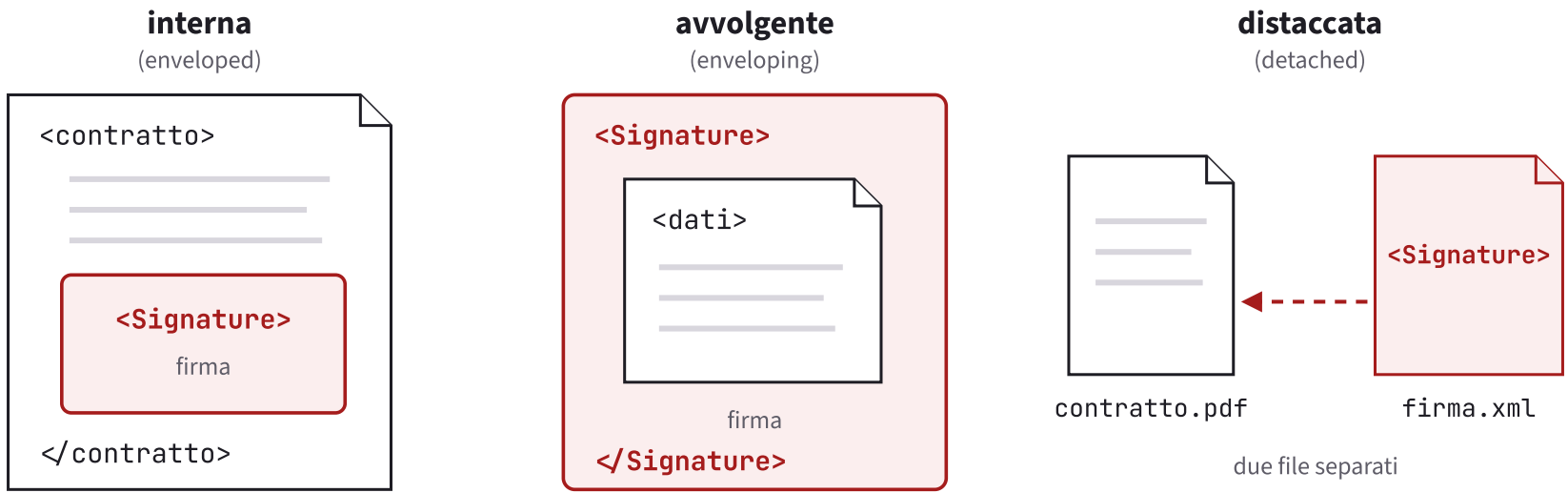
Reg. es. (UE) 2026/248, dal 23 febbraio 2027 (aggiunge JAdES e ASiC); abroga la Decisione 2015/1506 citata nel 2024.

ETSI

European Telecommunications Standards Institute: l'ente europeo di normazione che scrive gli standard tecnici di eIDAS.

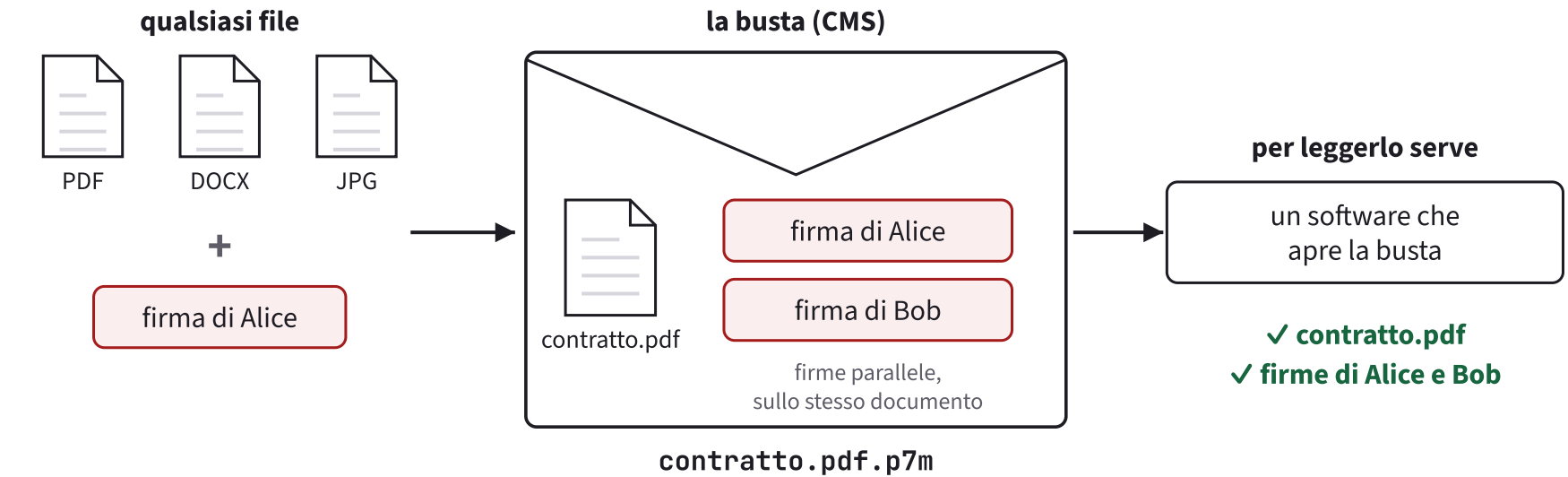
- Il regolamento dice **cosa** deve garantire una firma
- Le norme ETSI dicono **come**: formati, certificati, prestatori

XAdES: XML Advanced Electronic Signature

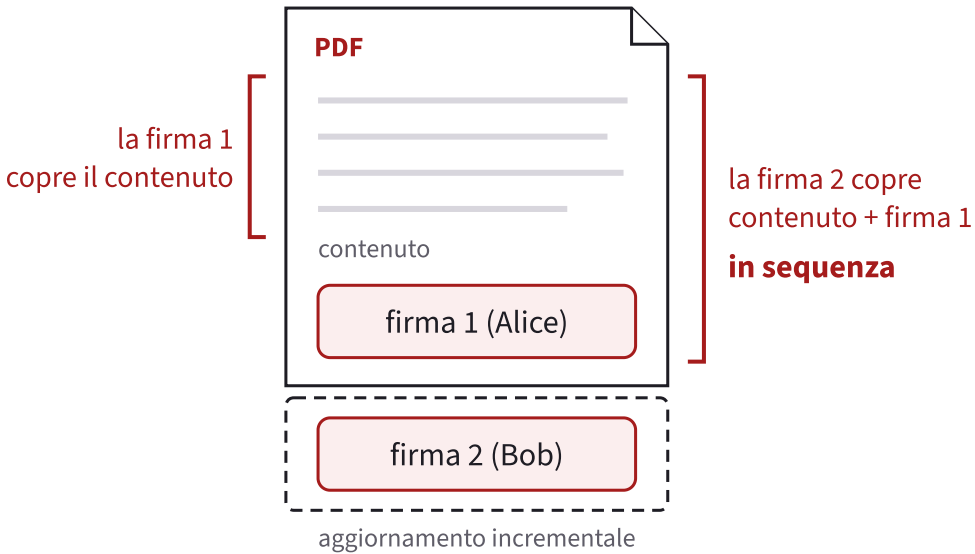


Firma in XML, leggibile da persone e programmi: comoda per l'elaborazione automatica e per più firmatari.

CAdES: CMS Advanced Electronic Signature



PAdES: PDF Advanced Electronic Signature





si apre con un normale lettore PDF

3. Firma elettronica qualificata (FEQ)



= firma autografa (art. 25, par. 2, eIDAS)

Le firme nel diritto italiano

Firma	Forma scritta ed efficacia probatoria
Semplice	valutate liberamente dal giudice
Avanzata	efficacia della scrittura privata (art. 2702 c.c.)
Qualificata / digitale	come sopra, e l'uso del dispositivo **si presume** del titolare

Per i trasferimenti di immobili e gli altri atti dell'art. 1350, nn. 1–12, c.c.: solo firma **qualificata o digitale**, a pena di nullità.

Che firma è?

	Semplice	Avanzata	Qualificata
Scrivo «Avv. Mario Rossi» in fondo a una email Dati elettronici usati per firmare, ma senza i requisiti dell'art. 26 eIDAS.	✓		
Clicco «Accetto» sulle condizioni di un sito Manifesta una volontà, ma non identifica in modo affidabile chi ha cliccato.	✓		
Firmo su un tablet in banca, con la penna elettronica La firma grafometrica è una FEA se il sistema rispetta le regole tecniche (art. 56 d.P.C.M. 22/2/2013); vale nei rapporti con la banca.		✓	
Firmo un atto con la smart card e ottengo un file .p7m Firma digitale: certificato qualificato e dispositivo sicuro (la smart card).			✓
Firmo da remoto con un codice OTP inviato sul telefono dal prestatore qualificato Firma remota: la chiave è custodita in un dispositivo qualificato del prestatore.			✓
Incollo nel PDF l'immagine scansionata della mia firma Un'immagine si copia e si incolla: nessun legame con il contenuto del documento.	✓		

Password monouso e verifica in due passaggi



1 · qualcosa che sai



2 · qualcosa che hai

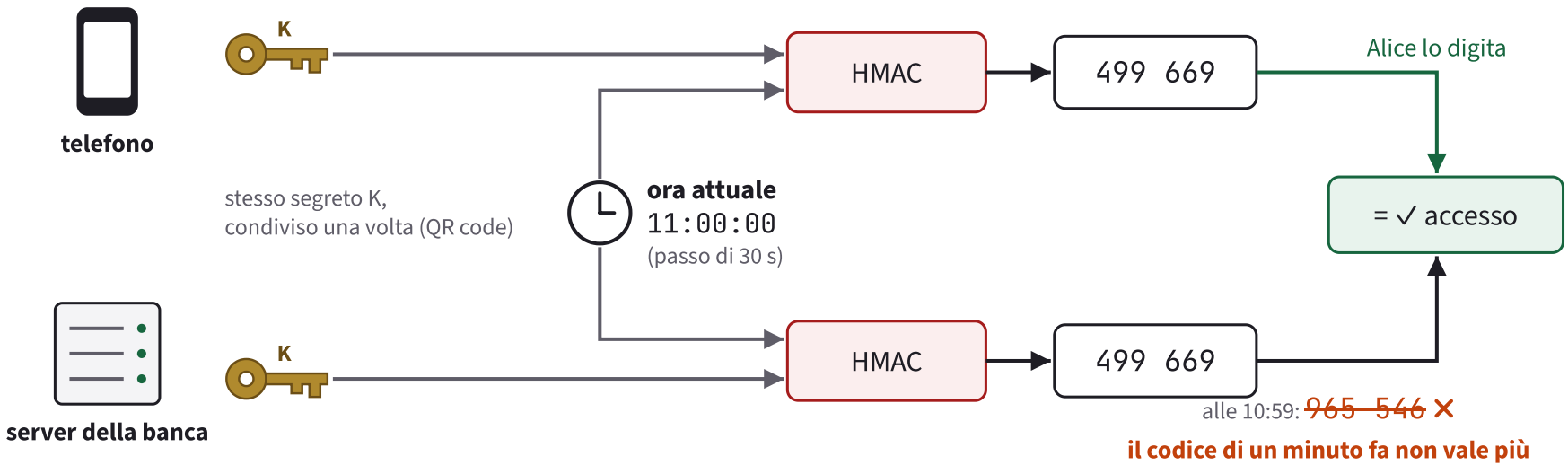
✓ accesso consentito

due fattori di categorie diverse

la sola password rubata non basta

Due fattori di categorie diverse: **qualcosa che sai**, **che hai**, **che sei**.

Il codice che cambia ogni 30 secondi



Proviamo: il codice sul telefono

Segreto condiviso, scambiato una sola volta con il QR code: `KRSX G5CT MVRX EZLU KRSX G5CT MVRX EZLU`

Il tuo telefono

499 669

nuovo codice tra 30 s

`codice = HMAC-SHA1(segreto, ora ÷ 30 s)`
ridotto a 6 cifre (RFC 6238)
intervallo n. 59.824.560

Il server della banca

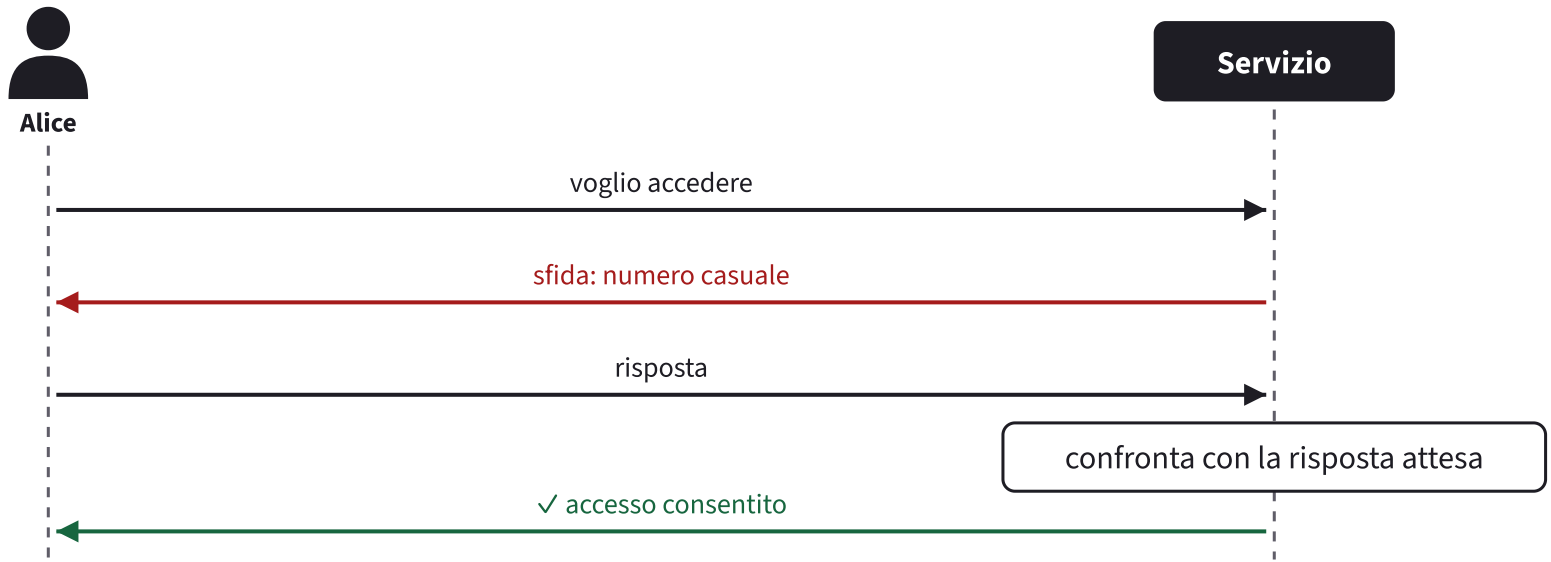
499 669

✓ **i due codici coincidono, senza che nessuno li trasmetta**

Il telefono invia 499 669: ✓ **codice accettato**

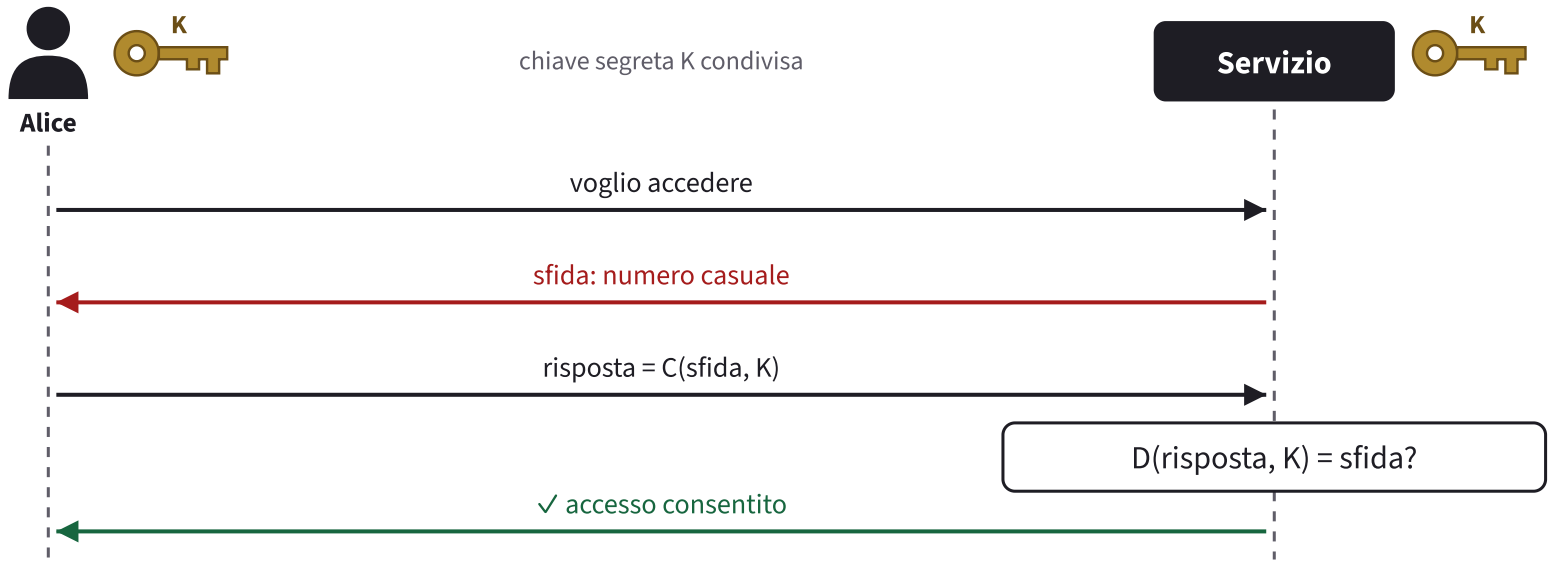
Codice di un minuto fa, 965 546: ✕ **codice scaduto**

Password monouso: sfida e risposta

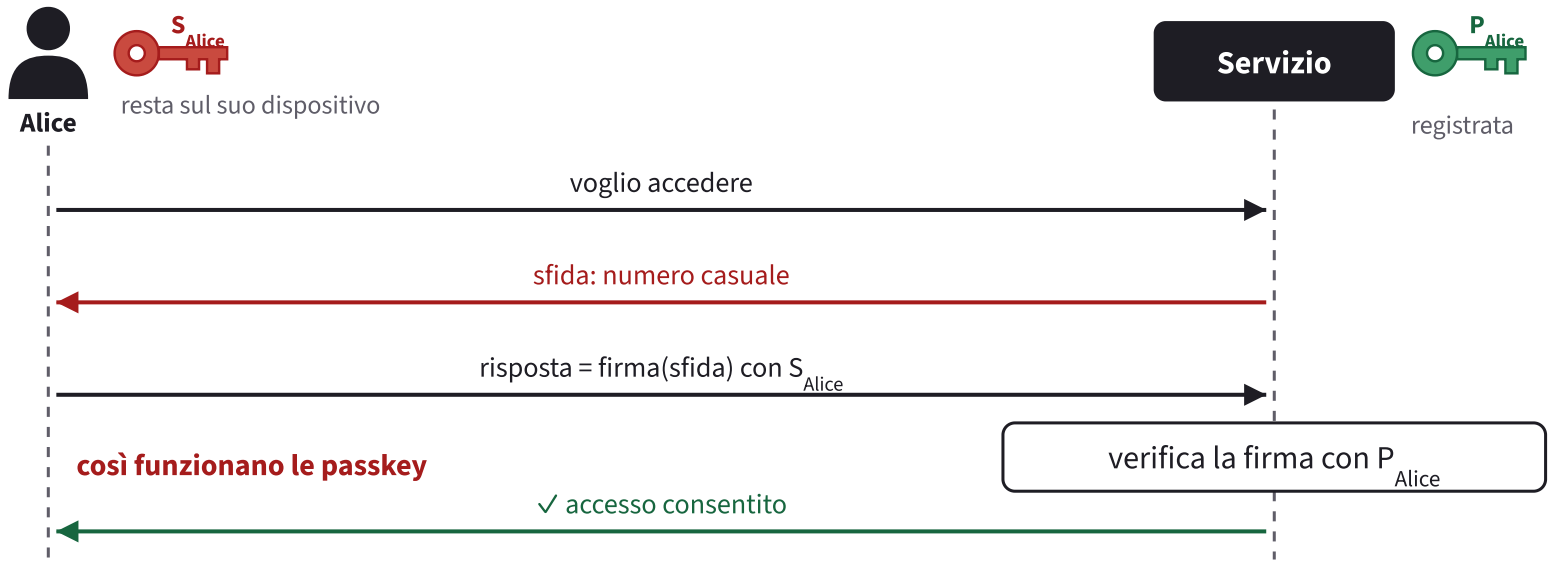


La risposta vale solo per quella sfida, e la sfida cambia ogni volta: registrarla non serve.

Sfida e risposta con crittografia simmetrica



Sfida e risposta con crittografia asimmetrica



È il meccanismo delle **passkey**, che stanno sostituendo le password.

Proviamo: accedere con una passkey

Prima: Alice ha registrato presso il Servizio la sua **chiave pubblica**; la privata resta sul suo dispositivo.

Alice	Servizio
1 Alice dichiara di voler accedere → «sono Alice»	
	2 Il Servizio invia una sfida casuale ← sfida: `ffd959636097cfe5`
3 Alice firma la sfida con la sua chiave privata → firma: `5dafe415f2b9…f3f6ff`	
	4 Il Servizio verifica la firma con la chiave pubblica di Alice ✓ **accesso consentito**

È il principio delle passkey (FIDO2/WebAuthn): niente segreti condivisi da rubare.

Verifica rapida

Un contratto di compravendita di un immobile, stipulato come documento informatico, quale firma richiede a pena di nullità?

- A Basta una firma semplice
- B Una firma avanzata
- **C Una firma qualificata o digitale** ✓ giusta

Per le scritture dell'art. 1350, nn. 1–12, c.c. l'art. 21, c. 2-bis, CAD richiede la firma qualificata o digitale (salvo sottoscrizione autenticata).

Perché un codice OTP rubato ieri non permette di entrare oggi?

- A Perché è cifrato
- **B Perché dipende dall'ora o da una sfida che nel frattempo è cambiata** ✓ giusta
- C Perché la banca lo cancella

Il codice è calcolato dall'ora corrente (TOTP) o da una sfida nuova: vale una sola volta e per poco tempo.

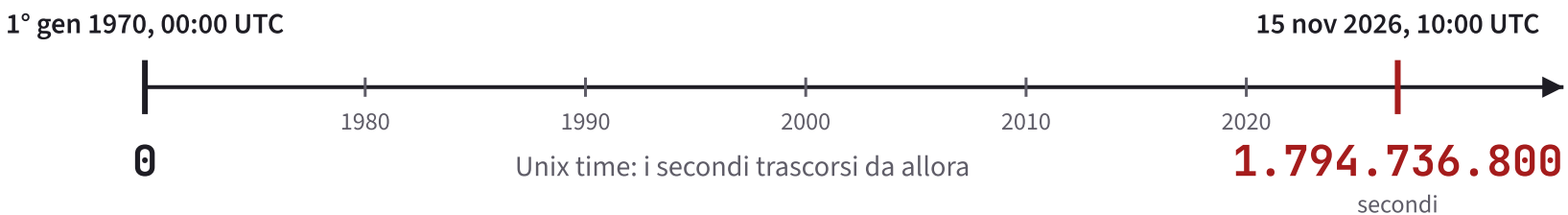
Capo VIII

VIII | Marcatura temporale fidata

Dimostrare che un documento esisteva a una certa data

Marca temporale (*timestamp*)

Data e ora scritte in caratteri: un **riferimento temporale**, che diventa **validazione temporale** quando un terzo lo lega a certi dati.



Lo stesso istante in ISO 8601
come lo scrivono i protocolli



Proviamo: che ore sono per un computer?

Tempo coordinato universale (UTC), formato ISO 8601 **2026-11-15T10:00:00Z**

Ora italiana (Europe/Rome), con lo scarto da UTC **2026-11-15T11:00:00+01:00**

Tempo Unix: secondi trascorsi dal 1° gennaio 1970, 00:00 UTC **1794736800**

Da tempo Unix a data

1602086059 | Esempio: 1602086059

UTC 2020-10-07T15:54:19Z
in Italia 2020-10-07T17:54:19+02:00

Da data e ora italiane a tempo Unix

15/11/2026, 11:00:00 | Adesso

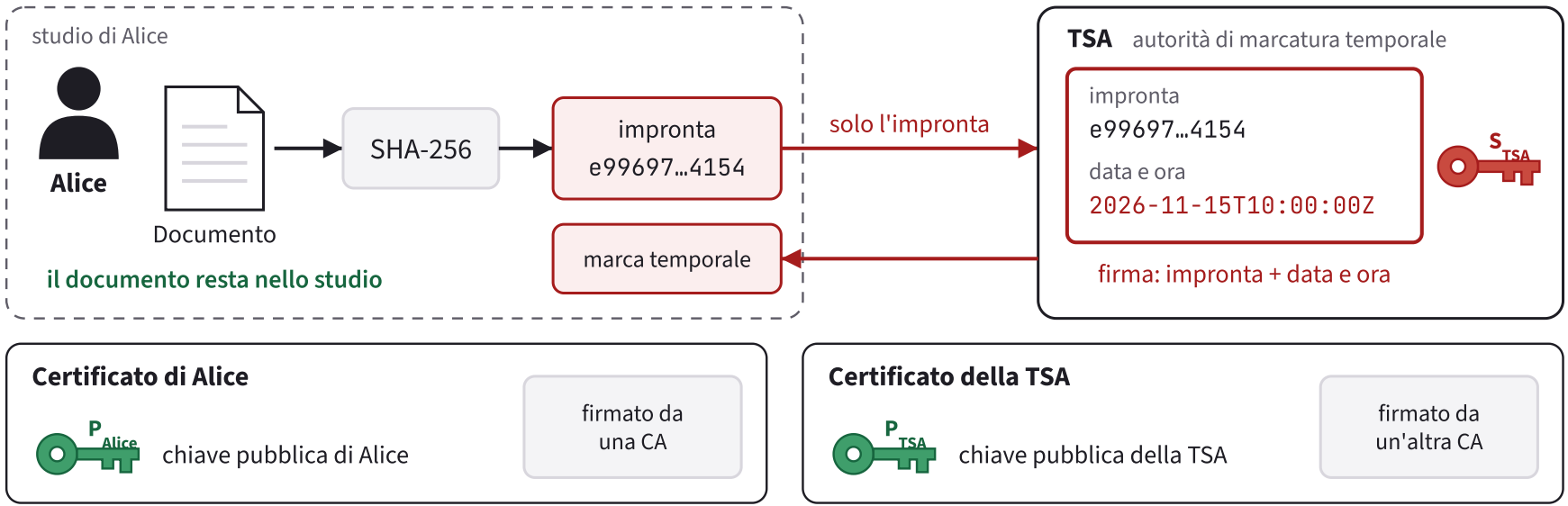
Unix 1794736800
UTC 2026-11-15T10:00:00Z

Marcatura temporale con infrastruttura a chiave pubblica

Un'autorità di marcatura temporale (TSA) attesta con la sua firma che un'impronta esisteva in un certo istante.

- Prova che i dati **esistevano** e **non sono cambiati** dopo; non prova chi ne è l'autore
- Protocollo standard: **RFC 3161**, su una PKI X.509
- La TSA riceve **solo l'impronta**: non vede mai il documento

La marcatura temporale, passo per passo



Proviamo: chiedere una marca temporale

1 Lo studio calcola l'impronta del documento

Parere legale sul contratto di fornitura Rossi–Bianchi, versione definitiva.

Impronta SHA-256: `e99697b0589c6177274b17102f72de7a800d56ce3a66d85a214a1007883b4154`

2 Invia alla TSA solo l'impronta: il documento resta nello studio

`e99697…4154`



3 La TSA aggiunge data e ora e firma impronta + data/ora

Impronta	`e99697b058…3b4154`
Data e ora	`2026-11-15T10:00:00Z`
Numero di serie	`2026-004817`
Firma della TSA	`613250e3b2…ab9944`

4 Chiunque può verificarla con il certificato della TSA

✓ marca temporale valida: questo documento esisteva già il 2026-11-15T10:00:00Z

Il pacchetto finale



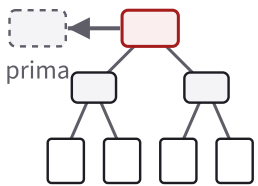
Lo standard ANSI ASC X9.95 e gli altri schemi

Oltre all'RFC 3161 esistono altri standard (ANSI X9.95 per la finanza, ISO/IEC 18014) e altre famiglie di schemi.



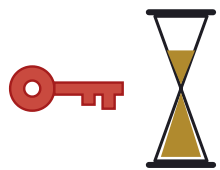
Basati su firma

la TSA firma
(PKI, RFC 3161)



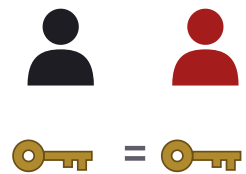
Collegati

ogni marca è legata
alle precedenti



Chiave transitoria

chiavi di firma
di breve durata



MAC

una chiave segreta
condivisa con la TSA



Database

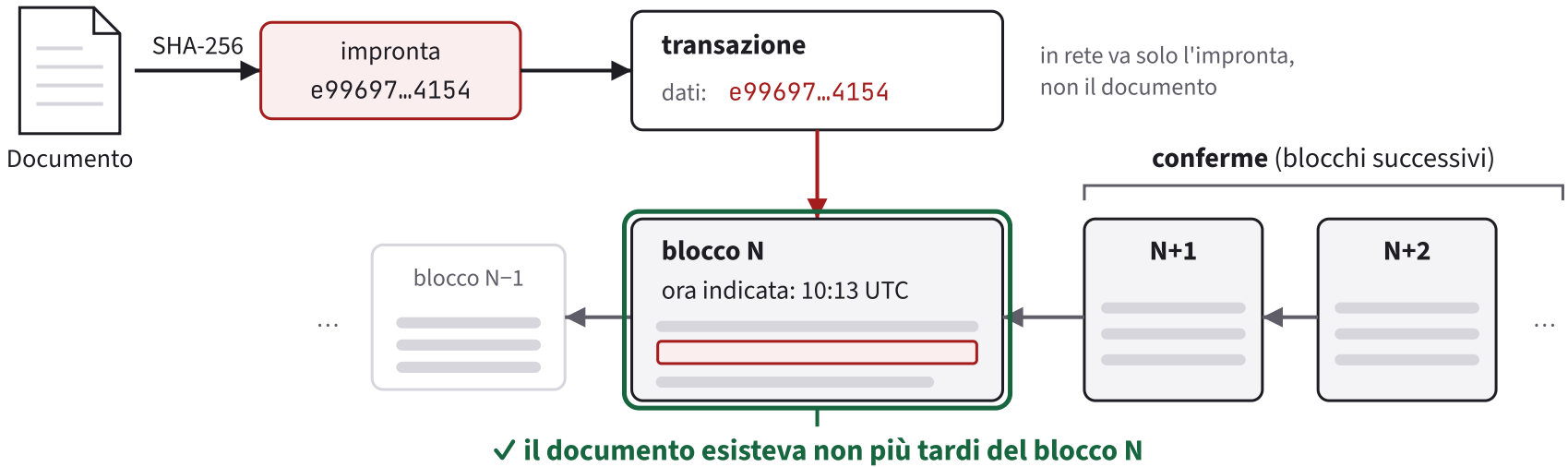
impronte in un
archivio fidato



Distribuiti

impronte in blockchain
nessuna TSA

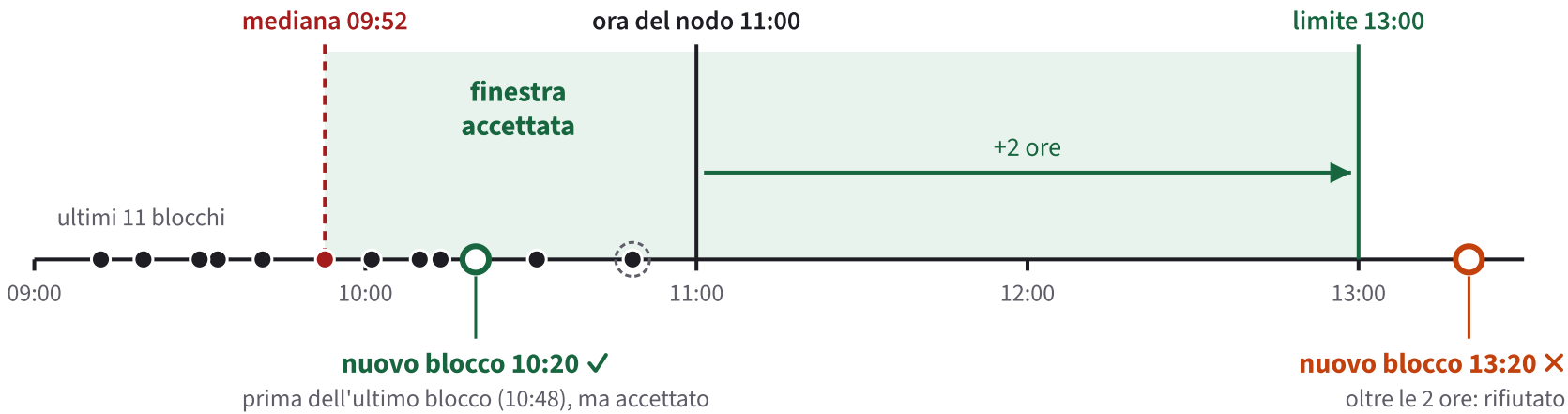
Marcatura temporale distribuita



Strumento gratuito: **OpenTimestamps** calcola l'impronta nel browser e la ancora nella blockchain di Bitcoin.

Marcatura temporale distribuita: i problemi

- Riscrivere la storia richiede la maggioranza della potenza di calcolo della rete (*proof of work*)
- Ma l'orario del blocco lo sceglie chi lo crea: è **approssimativo**

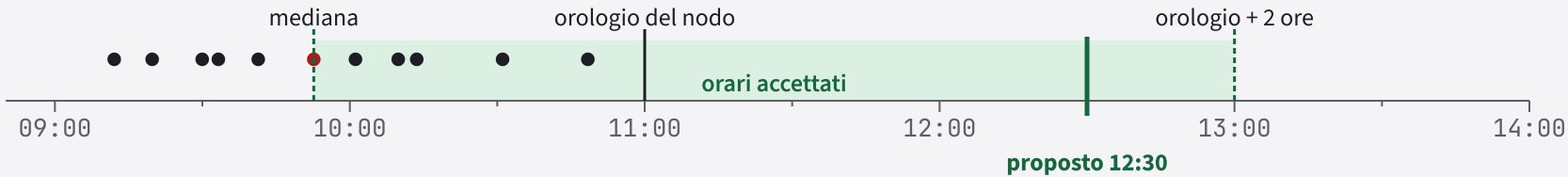


Proviamo: quale orario accetta la rete?

Orari degli ultimi 11 blocchi (UTC), nell'ordine della catena:

`09:12:04` `09:19:47` `09:33:15` `09:29:58` `09:41:22` **`09:52:40`** `10:01:09` `10:13:37` `10:09:51` `10:31:05` `10:48:26`

mediana degli ultimi 11 blocchi: **`09:52:40`** anteriore al blocco precedente Orologio del nodo: **`11:00:00`**



Orario scritto nel nuovo blocco: **`12:30 UTC`**

✓ accettato (anche se è 1 h 30 min nel futuro)

Quindi un blocco può indicare un orario fino a 2 ore nel futuro, o anteriore a blocchi già accettati.

Verifica rapida

Per ottenere una marca temporale su un parere riservato, cosa invia lo studio alla TSA?

A Il parere in PDF

B Solo l’impronta del parere ✓ giusta

C Il parere cifrato con la chiave pubblica della TSA

La TSA riceve e firma solo l’impronta con la data e l’ora: il contenuto del parere resta nello studio.

L’orario scritto in un blocco Bitcoin è preciso al secondo?

A Sì, è garantito dal consenso

B No: può scostarsi anche di un paio d’ore ✓ giusta

C Dipende dal fuso orario del notaio

Il protocollo accetta un blocco se il suo orario è successivo alla mediana degli 11 blocchi precedenti e non oltre 2 ore nel futuro.

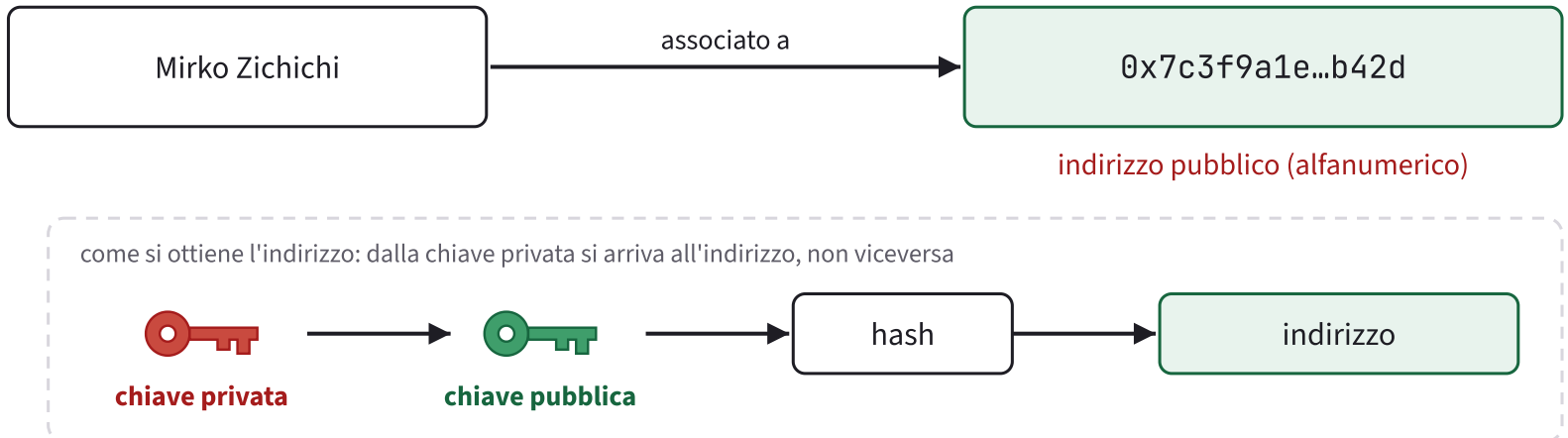
Capo IX

IX

Identità decentralizzate

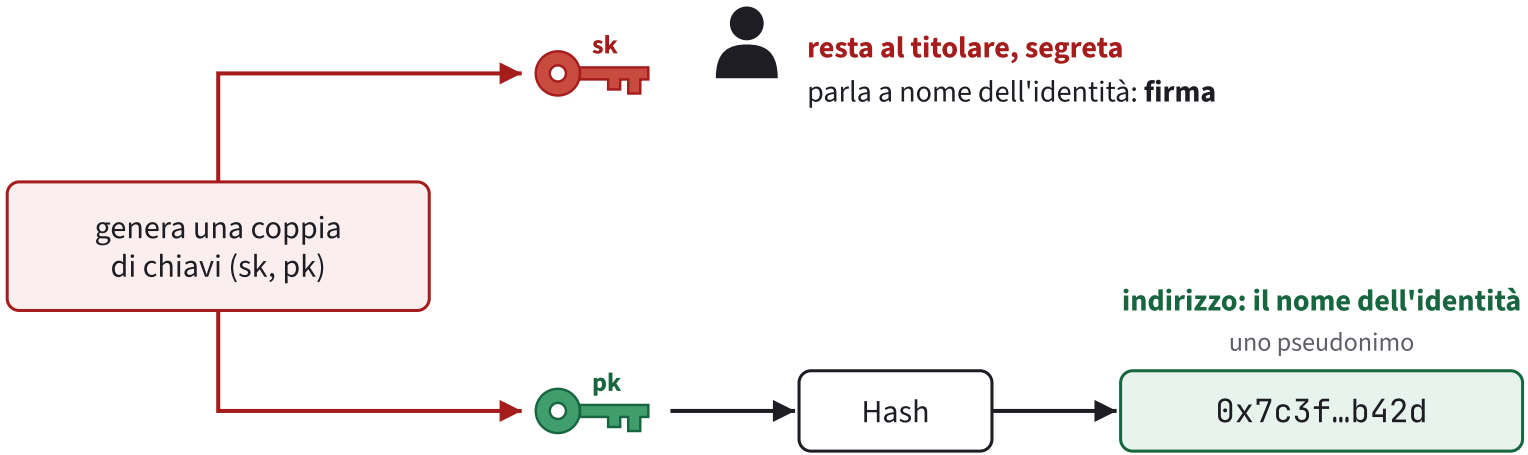
Quando la chiave pubblica diventa un nome

Le chiavi come identità



Su una blockchain non ci si registra con nome e cognome: si è un indirizzo.

Come si crea una nuova identità



nessuna autorità: chiunque può crearne quante vuole

Gestione decentralizzata delle identità

- Chiunque crea quante identità vuole, quando vuole
- Due chiavi uguali? Probabilità trascurabile, se il generatore casuale è buono
- Nessuna autorità che registri le identità
- In Bitcoin si chiamano **indirizzi**

Un indirizzo, nella pratica



il destinatario è solo un indirizzo: nessun nome

Proviamo: creare un'identità

Crea una nuova identità *Nessun registro centrale: chiunque può crearne quante vuole.* 3 create

	Indirizzo (impronta della chiave pubblica)	**Chiave pubblica**	**Chiave privata**
◉	`0x70f4675c06b9b25486f7b518071804bd8a9ba65c`	`04c961bd…a61d5d`	Mostra la chiave privata (nella realtà, mai!)
○	`0x36717af831a565ea1f1a270ef41be189fb692aae`	`047c3f35…dd3e2d`	Mostra la chiave privata (nella realtà, mai!)
○	`0xd3f2eb78c6e764c9c40e489f13d2305283641a9e`	`0417d39a…e00d64`	Mostra la chiave privata (nella realtà, mai!)

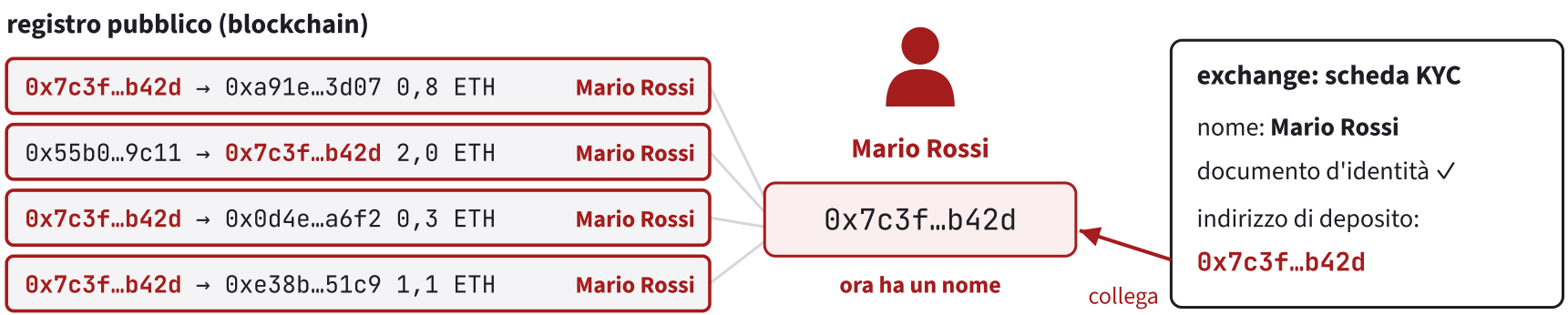
Messaggio

Dichiaro di essere il titolare di questo indirizzo.

Firma un messaggio con l'identità selezionata

Firma: `888d38f9a332…a4695a` ✓ **firma valida: il messaggio viene da chi controlla 0x70f467…a65c**

Pseudonimo non vuol dire anonimo



pseudonimo ≠ anonimo

basta un collegamento e tutte le transazioni dell'indirizzo, passate e future, sono sue

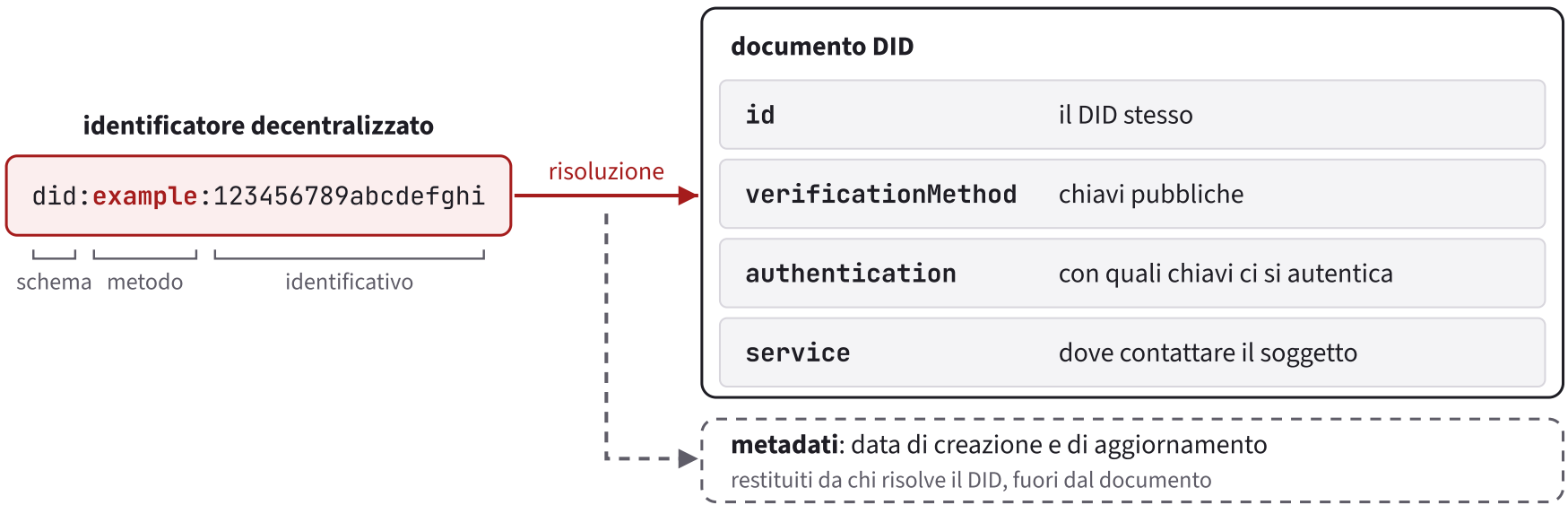
Identificatori decentralizzati (DID)

Un identificatore che il titolare controlla da sé, con le proprie chiavi.

- Identifica qualsiasi soggetto: persona, organizzazione, cosa
- Il titolare dimostra il controllo firmando con la chiave privata
- Nessun registro centrale; non serve nemmeno una blockchain

```
did:example:123456789abcdefghi
```

Il documento DID



Il documento DID, com'è scritto

```
{
  "@context": "https://www.w3.org/ns/did/v1",
  "id": "did:example:123456789abcdefghi",
  "verificationMethod": [{
    "id": "did:example:123456789abcdefghi#chiave-1",
    "type": "Multikey",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyMultibase": "z6MkmM42vxfqZQsv4ehtTjFF…"
  }],
  "authentication": ["#chiave-1"],
  "service": [{
    "id": "#sito",
    "type": "LinkedDomains",
    "serviceEndpoint": "https://studio-legale-esempio.it"
  }]
}
```

Schema: w3.org/TR/did-core

Verifica rapida

Un indirizzo Bitcoin non contiene nome e cognome. I dati delle sue transazioni sono dati personali?

- **A** No, sono anonimi
- **B** Possono esserlo: sono pseudonimi, riconducibili a una persona con informazioni ulteriori — ✓ **giusta**
- **C** Solo se l'importo supera 1.000 €

> Pseudonimo non è anonimo: se il titolare può essere identificato con mezzi ragionevolmente utilizzabili, si applica il GDPR.

Chi rilascia un DID?

- **A** Il Comune di residenza
- **B** Un'autorità di certificazione europea
- **C** Lo crea il titolare stesso, generando una coppia di chiavi — ✓ **giusta**

> Nessuna autorità centrale: il controllo del DID deriva dal possesso della chiave privata.

Grazie!

Domande?

Mirko Zichichi · mirko.zichichi@iota.org · mirkozichichi.me

Le slide interattive:
mirkozichichi.me/slides/sicurezzagiur2026

A

Allegato A

Appendice

Per approfondire: norme, standard, strumenti e un glossario

Per approfondire: sicurezza e crittografia (Capi I–IV)

- **NIST CSRC Glossary**: definizioni ufficiali di riservatezza, integrità, disponibilità, non ripudio, con la fonte di ciascuna. csrc.nist.gov/glossary
- **B. Schneier, *Secrets and Lies*, prefazione**: perché la sicurezza è un processo, non un prodotto. schneier.com/books/secrets-and-lies-pref
- **C. Mc Mahon, *In Defence of the Human Factor*** (2020): critica del luogo comune "l'utente è l'anello debole". pmc.ncbi.nlm.nih.gov/articles/PMC7365867
- **A. Kerckhoffs, *La cryptographie militaire*** (1883), il testo originale. petitcolas.net/kerckhoffs
- **NIST SP 800-63B-4** (2025): le regole attuali su password e autenticazione a più fattori. doi.org/10.6028/NIST.SP.800-63b-4
- **Menezes, van Oorschot, Vanstone, *Handbook of Applied Cryptography***, cap. 1, gratuito. cacr.uwaterloo.ca/hac
- **NIST IR 8547** (ottobre 2026): il calendario per abbandonare RSA e curve ellittiche. doi.org/10.6028/NIST.IR.8547
- **UE, tabella di marcia per la crittografia post-quantistica** (2025). digital-strategy.ec.europa.eu
- **Provare un cifrario**: CyberChef e cryptii.com

Per approfondire: hash e strutture dati (Capi V–VI)

- **A. Narayanan et al., *Bitcoin and Cryptocurrency Technologies*** (2016), cap. 1: hash, occultamento, puzzle, alberi di Merkle, identità. bitcoinbook.cs.princeton.edu
- **ACN, Linee guida funzioni crittografiche**: funzioni di hash e conservazione delle password, in italiano. acn.gov.it
- **NIST FIPS 180-4 (SHA-2) e FIPS 202 (SHA-3)**: gli standard delle funzioni hash raccomandate. csrc.nist.gov/pubs/fips/180-4/upd1/final
- **SHAttered** (2017): due PDF diversi con la stessa impronta SHA-1. shattered.io
- **NIST ritira SHA-1** (dicembre 2022): come e quando un algoritmo viene dismesso. nist.gov
- **Codice di procedura penale**, artt. 247, 254-bis e 260 (copia forense dei dati). normattiva.it
- **EDPB, Linee guida 02/2025 su blockchain e dati personali** (v2.0, luglio 2026). edpb.europa.eu

Per approfondire: firme elettroniche (Capo VII)

- **Regolamento eIDAS**, testo consolidato in italiano. eur-lex.europa.eu
- **Reg. (UE) 2024/1183 (eIDAS 2)** e portafoglio europeo di identità digitale. eur-lex.europa.eu
- **Reg. di esecuzione (UE) 2026/248**: i formati di firma avanzata che la PA deve accettare. eur-lex.europa.eu
- **CAD**, art. 20 (valore probatorio) e art. 21 (forma e firme). normattiva.it
- **AgID, IT-Wallet**: base normativa e linee guida. agid.gov.it/it/piattaforme/it-wallet
- **Garante privacy, provvedimento sulla biometria** (2014), anche sulla firma grafometrica. garanteprivacy.it
- **Commissione UE, DSS**: firmare e validare firme eIDAS. ec.europa.eu/digital-building-blocks/DSS/webapp-demo
- **Passkey**: W3C Web Authentication Level 3 e FIDO Alliance. w3.org/TR/webauthn-3, fidoalliance.org/passkeys

Per approfondire: tempo e identità (Capi VIII–IX)

- **IETF RFC 3161, *Time-Stamp Protocol***: la TSA riceve solo l'impronta. rfc-editor.org/rfc/rfc3161
- **Art. 8-ter d.l. 135/2018**: registri distribuiti, smart contract e validazione temporale. normattiva.it
- **eIDAS 2**: validazione temporale (artt. 41–42) e registri elettronici qualificati (artt. 45 duodecies e 45 terdecies). eur-lex.europa.eu
- **OpenTimestamps**: una marca temporale su Bitcoin, gratis, dal browser. opentimestamps.org
- **Corte di giustizia UE, C-413/23 P, GEPD c. SRB** (2025): quando i dati pseudonimizzati sono personali. eur-lex.europa.eu
- **W3C, *Decentralized Identifiers (DIDs) v1.0***. w3.org/TR/did-core
- **EUDI Wallet, Architecture and Reference Framework**: formati e protocolli del portafoglio europeo. eudi.dev

Le notizie citate in apertura

- **Rapporto Clusit 2026**, presentazione del 17 marzo 2026. securitysummit.it
- **ACN, Relazione annuale 2025** (sintesi). puntosicuro.it
- **Frigoriferi dei supermercati**, Il Post, 3 novembre 2026. ilpost.it
- **Una settimana di attacchi hacker e allarmi bomba**, Il Post, 9 novembre 2026. ilpost.it
- **Presunti file della DIA**, AGI, 9 novembre 2026. agi.it
- **Caso Paragon**, Internazionale, 12 marzo 2026. internazionale.it
- **La voce clonata del ministro Crosetto**, Il Post, 11 febbraio 2025. ilpost.it
- **Record di fattorizzazione RSA-260**, Cognition, 9 settembre 2026. cognition.com
- **Normativa**: l. 90/2024, d.lgs. 138/2024 (NIS2), Cyber Resilience Act

Glossario (1/4)

Termine	In inglese	In breve
Riservatezza	*confidentiality*	solo chi è autorizzato accede ai dati
Integrità	*integrity*	i dati non sono modificati senza autorizzazione
Disponibilità	*availability*	dati e servizi sono accessibili quando servono
Autenticazione	*authentication*	verificare l'identità di chi comunica
Non ripudio	*non-repudiation*	l'autore non può negare in seguito un'azione
Testo in chiaro / cifrato	*plaintext / ciphertext*	il messaggio prima e dopo la cifratura
Crittografia simmetrica	*secret-key cryptography*	la stessa chiave cifra e decifra
Crittografia asimmetrica	*public-key cryptography*	chiave pubblica e chiave privata, diverse e collegate
Attacco a forza bruta	*brute-force attack*	provare tutte le chiavi o password possibili
Attacco a dizionario	*dictionary attack*	provare prima le password più comuni

Glossario (2/4)

Termine	In inglese	In breve
Impronta	*hash, digest*	valore di lunghezza fissa calcolato da un dato
Collisione	*collision*	due dati diversi con la stessa impronta
Nonce	*nonce*	numero casuale usato una sola volta
Impegno	*commitment*	vincolarsi a un valore senza rivelarlo subito
Puntatore hash	*hash pointer*	riferimento a un dato insieme alla sua impronta
Albero di Merkle	*Merkle tree*	albero di impronte riassunto da un'unica radice
Firma digitale	*digital signature*	impronta cifrata con la chiave privata del firmatario

Glossario (3/4)

Termine	In inglese	In breve
Certificato	*certificate*	documento che lega una chiave pubblica a un'identità
Autorità di certificazione	*certification authority (CA)*	soggetto fidato che firma i certificati
Prestatore di servizi fiduciari qualificato	*qualified trust service provider (QTSP)*	soggetto vigilato che rilascia certificati qualificati, marche temporali e altri servizi eIDAS
Dispositivo per la firma qualificata	*QSCD*	smart card, token o sistema remoto certificato che custodisce la chiave privata
Infrastruttura a chiave pubblica	*PKI*	regole, autorità e sistemi che gestiscono i certificati
Marca temporale	*time stamp*	attestazione firmata di data e ora di un'impronta
Autorità di marcatura temporale	*time stamping authority (TSA)*	soggetto che rilascia le marche temporali

Glossario (4/4)

Termine	In inglese	In breve
Registro distribuito	*distributed ledger (DLT)*	registro replicato tra molti nodi che si accordano tra loro
Identificatore decentralizzato	*decentralized identifier (DID)*	identificatore controllato dal titolare tramite chiavi
Autenticazione a più fattori	*multi-factor authentication (MFA)*	almeno due tra: qualcosa che sai, che hai, che sei
Password monouso	*one-time password (OTP)*	codice valido una sola volta o per pochi secondi
Passkey	*passkey (FIDO2/WebAuthn)*	accesso senza password basato su una coppia di chiavi
Phishing	*phishing*	messaggio ingannevole per carpire credenziali o denaro
Ransomware	*ransomware*	software che cifra i dati della vittima e chiede un riscatto
Attacco DDoS	*distributed denial of service*	sommergere un servizio di richieste per renderlo irraggiungibile

Crediti

- **Immagini**: cilindro di Jefferson, foto NSA, pubblico dominio; Enigma al Museo Nazionale della Scienza e della Tecnologia "Leonardo da Vinci" di Milano, foto di Alessandro Nassiri, CC BY-SA 4.0; ritratto di Auguste Kerckhoffs, pubblico dominio (tutte da Wikimedia Commons)
- **Contenuti**: slide in collaborazione con prof. Stefano Ferretti; edizione interattiva aggiornata a novembre 2026
- **Le dimostrazioni** girano interamente nel browser: nessun dato inserito viene inviato a terzi

Versione interattiva delle slide:
mirkozichichi.me/slides/sicurezzagiur2026