UNIVERSITÀ POLITECNICA DELLE MARCHE

DII
Dipartimento di Ingegneria dell'Informazione

unimc

Algoritmi di crittografia

Funzionamenti e innovazioni

Esposito Francesco Italo

Francescoitaloesposito@gmail.com

Venerdì 12 Aprile 2024

Corso di Perfezionamento in Cybersecurity, Cyber Risk and Data Protection

Algoritmi di crittografia

Presentazione:

Il seguente elaborato ha lo scopo di fornire una panoramica sugli algoritmi crittografici, introducendo innanzitutto l'importanza della crittografia quale misura fondamentale di sicurezza dei dati personali. Tale rilevanza non è solo tecnica, ma anche giuridica: le normative europee in materia di privacy e data protection (come il GDPR, art. 5, 25 e 32) richiedono esplicitamente l'adozione di misure di sicurezza quali la cifratura.

Successivamente verranno analizzati alcuni algoritmi di riferimento, a partire da quelli più noti, come RSA e AES, per poi presentare soluzioni più recenti e innovative, come la crittografia a curve ellittiche (ECC), che oggi rappresentano lo standard in molte applicazioni di sicurezza. Presentando al termine una panoramica sugli algoritmi Post-Quantum

Privacy e sicurezza: cosa dice la legge

Negli ultimi anni la necessità di garantire un trattamento dei dati personali **più sicuro e rispettoso della privacy** è diventata un tema centrale nel dibattito pubblico e istituzionale. A confermare ciò non vi sono soltanto indicazioni di carattere tecnico o richieste informali, ma anche **precisi riferimenti normativi**, ai quali ogni organizzazione che gestisce dati (*data controller* o *data processor*) è tenuta a conformarsi.

- ✓ GDPR (General Data Protection Regulation/Regolamento UE n. 679/2016)
- ✓ Linee guide ENISA
- ✓ ISO/IEC 27001

Privacy e sicurezza: cosa dice la legge

- ⚖ Art.5 GDPR: I dati personali devono essere trattati secondo i principi di **integrità** e **riservatezza** (**CIA triad**).
- ⚖ Art. 25 GDPR (*privacy by design e by default*): Le misure di protezione (tra cui la crittografia) devono essere incorporate sin dalla progettazione dei sistemi.
- ⚖ Art. 32 GDPR (*sicurezza del trattamento*): I titolari del trattamento devono adottare misure tecniche adeguate e la **cifratura dei dati personali** ne è esempio diretto.
- ⚖ Direttiva NIS2 (2023): rafforza la sicurezza delle infrastrutture critiche e introduce obblighi su cifratura e gestione chiavi.
- ⚖ ISO/IEC 27001: standard internazionale sulla sicurezza delle informazioni che include la crittografia tra i controlli fondamentali.

Dati at rest, in transit, in use

Il GDPR, pur non utilizzando espressamente la terminologia tecnica 'at rest', 'in transit' e 'in use', richiama la necessità di proteggere i dati personali nelle diverse fasi del trattamento:

Art. 32, par. 2 GDPR

"Nel valutare l'adeguato livello di sicurezza si tiene conto in particolare dei rischi presentati dal trattamento che derivano, in particolare, dalla distruzione, dalla perdita, dalla modifica, dalla divulgazione non autorizzata o dall'accesso, in modo accidentale o illecito, a dati personali trasmessi, conservati o comunque trattati."

Stato dati	Descrizione	Rischi principali	Algoritmi raccomandati
Data at rest	Dati archiviati su dispositivi fisici (database, server, backup, hard disk)	- Accesso non autorizzato - Furto di dispositivi - Compromissione storage	AES-256 Crittografia simmetrica efficiente per grandi volumi
Data in transit	Dati in movimento attraverso reti durante trasmissioni o comunicazioni	- Intercettazione - Manipolazione durante trasferimento - Man-in-the-middle attacks	RSA/ECC + AES Handshake per scambio chiavi + cifratura veloce
Data in use	Dati attivamente elaborati da applicazioni, sistemi o utenti	- Accesso abusivo durante elaborazione - Memory dumps - Privilege escalation	RSA/ECC (firme digitali) Autenticazione e integrità + controlli di accesso

Consigli o doveri?

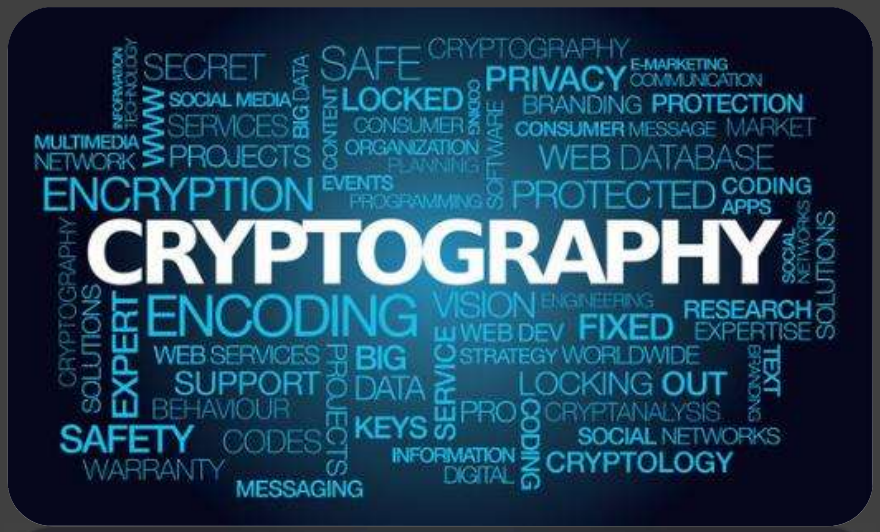
Best practice

Sono per lo più linee guida o raccomandazioni (es. ENISA, ISO). Le best practice diventano spesso “standard di riferimento” per valutare l’adeguatezza delle misure, non sono vincolanti per legge, ma rappresentano modelli consolidati di comportamento.

Obblighi di legge

Sono regole vincolanti e sanzionabili, che le organizzazioni devono rispettare (es. GDPR, NIS2). La non conformità può portare multe o responsabilità legale.

Le basi della crittografia



Cos'è la crittografia?

Tecnica per proteggere dati → trasformazione da *testo in chiaro* a *testo cifrato*.

Obiettivi principali:

- **Riservatezza** (solo i destinatari autorizzati leggono i dati).
- **Integrità** (i dati non vengono alterati).
- **Disponibilità** (garantisce che le informazioni siano accessibili agli utenti autorizzati quando necessario)
- **Autenticità** (certezza dell'identità del mittente).
- **Non ripudio** (il mittente non può negare di aver inviato i dati).

Tipi di crittografia

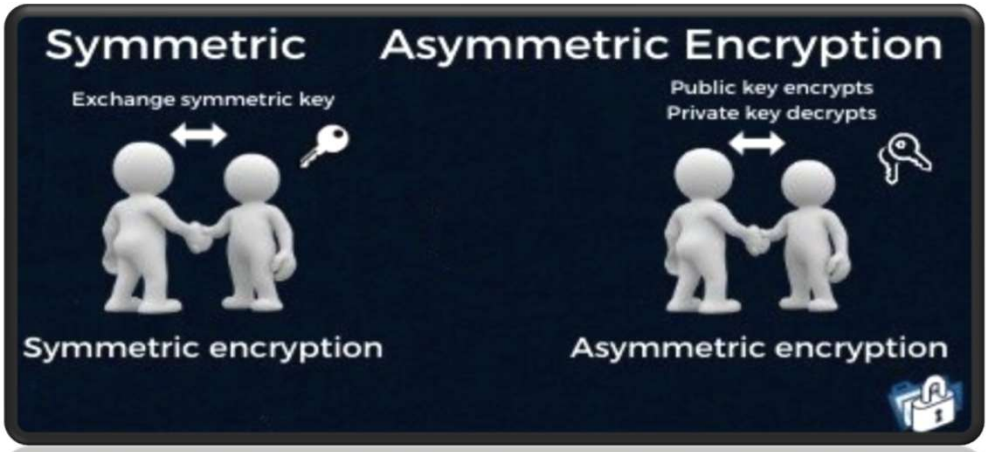
Nella crittografia simmetrica la stessa chiave viene utilizzata sia per cifrare che per decifrare i dati. È un metodo molto veloce ed efficiente, ideale per proteggere grandi quantità di informazioni, ma presenta una criticità: la chiave deve essere scambiata tra le parti in modo sicuro, altrimenti l'intero sistema risulta vulnerabile.

La crittografia asimmetrica utilizza una coppia di chiavi: una pubblica e una privata. La chiave pubblica può essere distribuita liberamente e serve per cifrare, mentre solo la chiave privata, custodita dal destinatario, permette di decifrare. Questo approccio rende più sicuro lo scambio delle chiavi, ma è meno efficiente per cifrare grandi volumi di dati.

Usate insieme: *asimmetrica per scambiare la chiave, la simmetrica per cifrare i dati*

EKM e EDM

Come lavorano insieme?



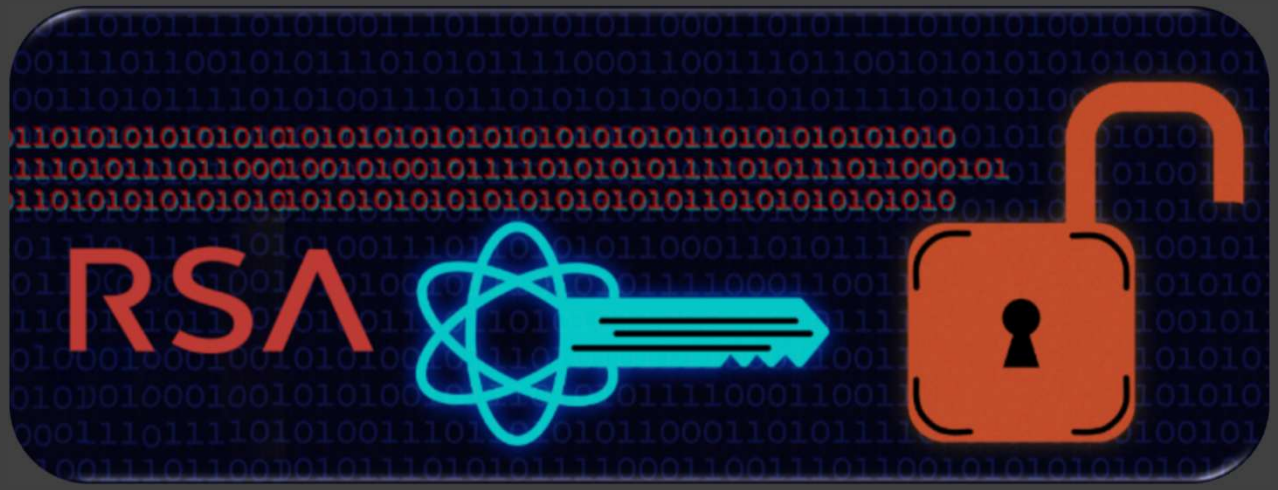
Nelle comunicazioni sicure, i due approcci crittografici vengono combinati per sfruttarne i vantaggi.

Con il **Key Encapsulation Mechanism (EKM)** si utilizza la crittografia asimmetrica (es. RSA o ECC) per cifrare e trasmettere in modo sicuro la chiave simmetrica. Una volta che entrambe le parti condividono la stessa chiave, entra in gioco il **Data Encapsulation Mechanism (EDM)**, che sfrutta un algoritmo simmetrico (es. AES) per cifrare i dati veri e propri, garantendo velocità ed efficienza.

La gestione delle chiavi

- La sicurezza non dipende solo dall'algoritmo, ma dalla forza e gestione delle chiavi.
  - Key lenght
  - Maggiore resistenza agli attacchi brute-force
  - Key management
  - Distribuzione, conservazione e revoca sicure.
  - Rotazione periodica e revoca in caso di compromissione.
  - Generazione sicura (randomicità)

Algoritmi asimmetrici : RSA



Introduzione e funzionamento di base

- Creato nel 1977 da **Rivest, Shamir e Adleman**.

▪Primo algoritmo a chiave pubblica di uso pratico.

▪Basato sulla difficoltà di fattorizzare grandi numeri primi.

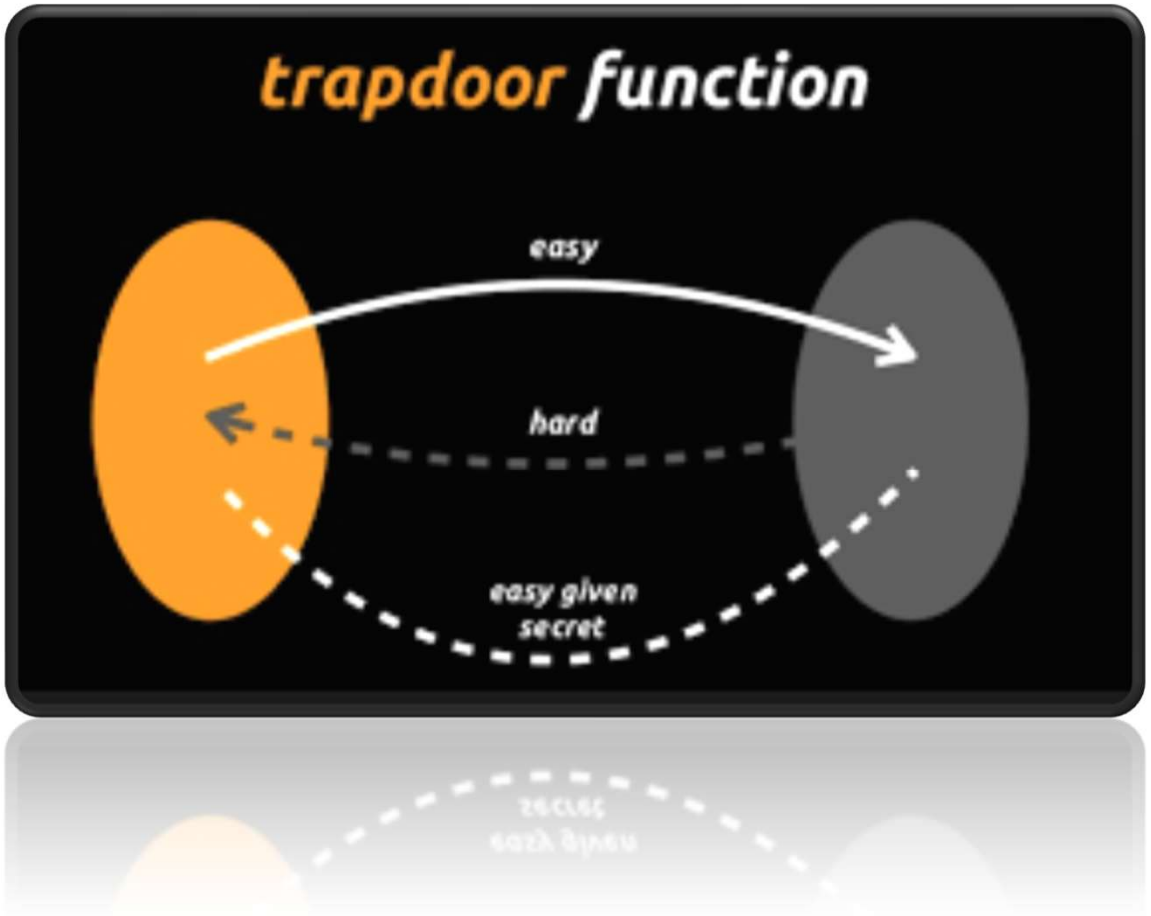
Principali utilizzi:

▪Scambio sicuro di chiavi.

▪Firme digitali.

La matematica dietro l'algoritmo

- L'uso principale della cifratura RSA è quello di **scambiare in modo sicuro una chiave** per un cifrario simmetrico (meccanismo di trasporto della chiave). Viene spesso utilizzato insieme a un cifrario simmetrico come AES, dove quest'ultimo si occupa della cifratura effettiva dei dati (EDM).
- La funzione unidirezionale su cui si basa RSA è il **problema della fattorizzazione di interi**: moltiplicare due numeri primi molto grandi è computazionalmente semplice, ma **fattorizzare il prodotto risultante è estremamente difficile**.
- Si fonda quindi su una **trapdoor one-way function**: elevamento modulare con esponente pubblico, invertibile solo conoscendo l'informazione segreta (fattori primi di n).



Applicazione pratica delle normative

Data at rest

Crittografia di chiavi simmetriche per protezione database e archivi

Esempi pratici: dati sanitari su server, documenti legali archiviati.

Dati in transit

Handshake TLS/SSL per stabilire connessioni sicure.

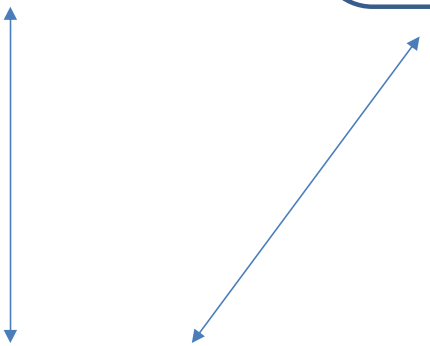
Applicazione GDPR: certificati HTTPS per siti web con dati personali.

Dati in use

Utilizzo limitato: firme digitali per autenticazione utenti

Funzione specifica: validazione transazioni e documenti durante l'elaborazione.

Nota importante: RSA per "data in use" ha limitazioni perché richiede decrittografia per l'elaborazione, cosa che lo rende meno adatto per questo scenario rispetto ad AES.



RSA garantisce l'infrastruttura crittografica per la conformità all'articolo 32 GDPR

I limiti di RSA

➢ Lentezza

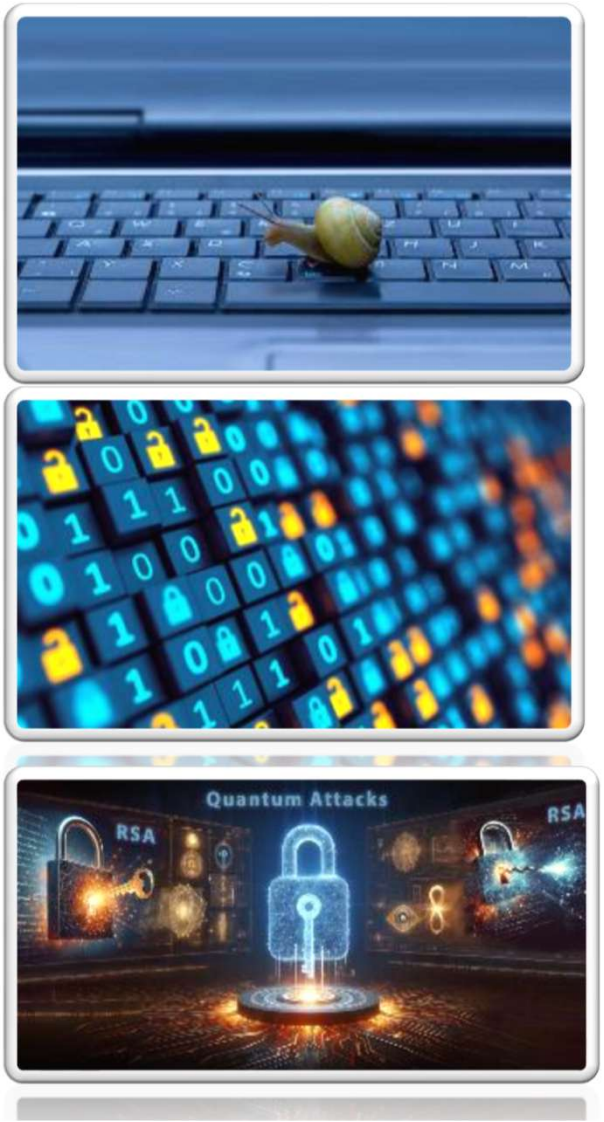
- RSA è lento perché le operazioni modulari sono computazionalmente molto costose, questo lo rende poco efficiente per lo scambio di dati, motivo per cui viene usato solo per lo scambio di chiavi.

➢ Chiavi troppo lunghe

- Standard attuali: 2048–4096 bit a confronto: ECC fornisce stessa sicurezza con chiavi molto più corte. es. RSA 3072 ≈ ECC 256 (stesso livello di sicurezza).

➢ La minaccia quantistica

- Algoritmo di Shor può fattorizzare n rapidamente su un computer quantistico.
  RSA perderebbe tutta la sua sicurezza
- Necessità di passare ad algoritmi post-quantum (Kyber, Dilithium)



RSA: pietra miliare, ma non il futuro

- Ha introdotto la **crittografia a chiave pubblica**, rivoluzionando la sicurezza informatica.
- Ancora oggi usato per **firma digitale** e **scambio chiavi**.
- **Limiti evidenti:** lentezza, chiavi molto lunghe, vulnerabilità a quantum computing.
  - Sta progressivamente lasciando spazio a soluzioni più moderne (ECC, algoritmi post-quantum).

RSA è fondamentale per capire il passato e il presente della crittografia, ma non sarà la tecnologia del futuro.

Algoritmi simmetrici: AES

Origine di AES

- Alla fine degli anni '90, il Data Encryption Standard (DES) non era più considerato sicuro: la lunghezza della chiave (56 bit) era troppo ridotta per resistere ad attacchi brute force con i computer dell'epoca.
- Nel 1997 il National Institute of Standards and Technology (NIST) avviò un concorso internazionale per definire il nuovo standard crittografico che avrebbe sostituito DES. Tra i vari algoritmi proposti, fu scelto nel 2001 l'algoritmo AES (crittografi belgi Joan Daemen e Vincent Rijmen).
- Garantiva un ottimo compromesso tra sicurezza e prestazioni. Era efficiente sia su hardware che su software. Supportava chiavi di diverse lunghezze (128, 192, 256 bit).

Come funziona?

- AES è un cifrario a blocchi: lavora su blocchi di 128 bit (16 byte) di plaintext, che vengono trasformati in blocchi di ciphertext della stessa dimensione.
- L'algoritmo applica una serie di trasformazioni ripetute, organizzate in round, il cui numero dipende dalla lunghezza della chiave:

  10 round per AES-128 12 round per AES-192 14 round per AES-256

- AES si basa su quattro trasformazioni principali l'ultima trasformazione, applicata dopo le altre manipolazioni, che rende effettivamente il blocco cifrato dipendente dalla chiave, garantendo la segretezza del dato.

Applicazione pratica delle normative

Data at rest

Crittografia simmetrica AES-256 per protezione di database, file system e backup

Esempi pratici: cartelle cliniche cifrate, archivi finanziari, documenti aziendali.

Rischi principali: Accesso non autorizzato, furto di dispositivi.

Dati in transit

Utilizzo di AES dopo handshake RSA/ECC per garantire trasmissioni veloci e sicure.

Dati in use

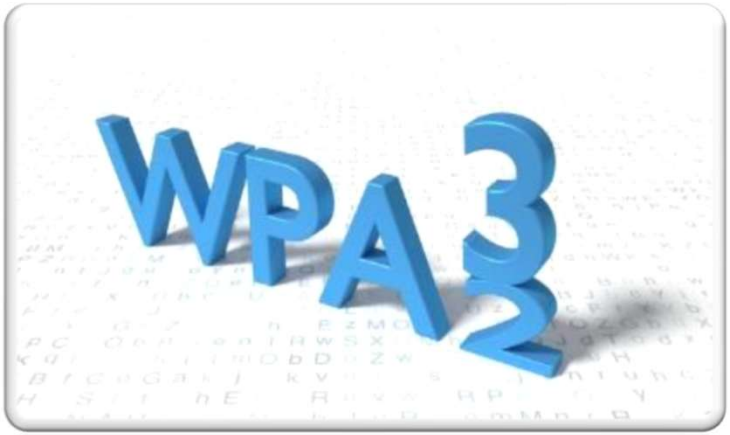
ECDSA/RSA: firme digitali per verifica autenticità e integrità.

AES è il principale standard simmetrico raccomandato per soddisfare i requisiti crittografici dell'Art. 32 GDPR

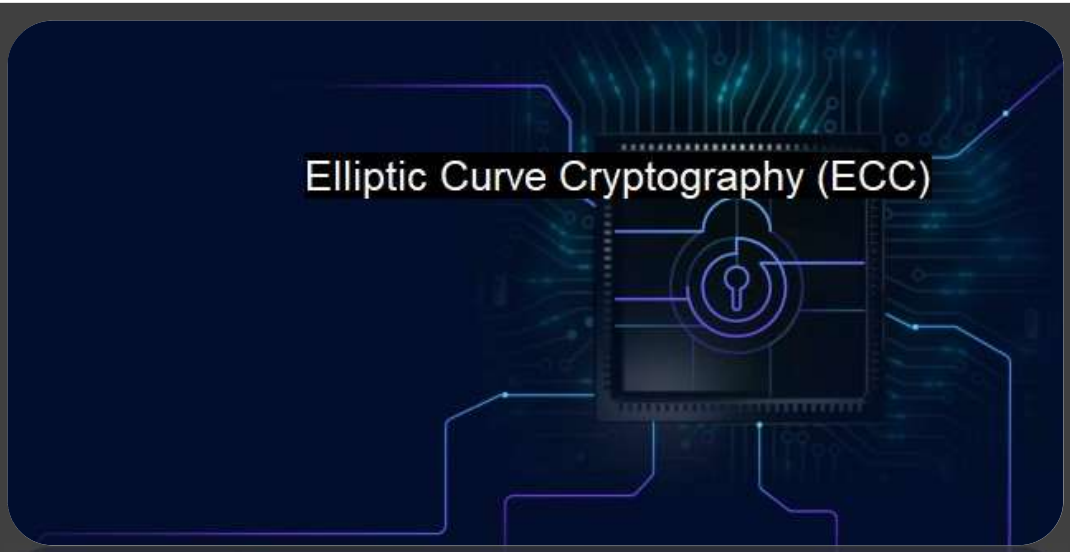
Dove troviamo AES?

Nonostante non sia recente, AES è ancora utilizzato in alcuni ambiti:

- Nei protocolli moderni come **TLS 1.2 e TLS 1.3**, AES (in modalità GCM, cioè AES-GCM) è ancora la scelta principale per cifrare il traffico HTTPS.
- In **VPN** (es. IPsec, OpenVPN, WireGuard), AES è ancora molto diffuso come algoritmo di cifratura.
- In **Wi-Fi**, lo standard **WPA2/WPA3** usa AES come base (AES-CCMP).
- Nella **messaggistica** (es. WhatsApp, Signal), AES viene usato come parte della cifratura simmetrica per proteggere i messaggi.



Algoritmi innovativi: ECC



Da RSA ad ECC: una nuova generazione

Come abbiamo visto, RSA diventa progressivamente lento e poco pratico man mano che le chiavi aumentano di lunghezza. Per questo motivo si è resa necessaria una nuova soluzione crittografica più efficiente.
Negli anni '80 viene proposto l'algoritmo a curve ellittiche (ECC), che dagli anni '90 in poi inizia a diffondersi e oggi rappresenta lo standard più utilizzato nei protocolli moderni.

Perché ECC?

- Sicurezza elevata con chiavi molto più corte:
  - RSA a 3072 bit ≈ sicurezza di ECC a 256 bit.
- Prestazioni migliori: meno memoria e meno calcoli → utile per dispositivi mobili e IoT.
- Flessibilità: può essere usato sia per cifrare, sia per firme digitali.
- È oggi uno standard raccomandato da NIST e adottato in TLS 1.3.

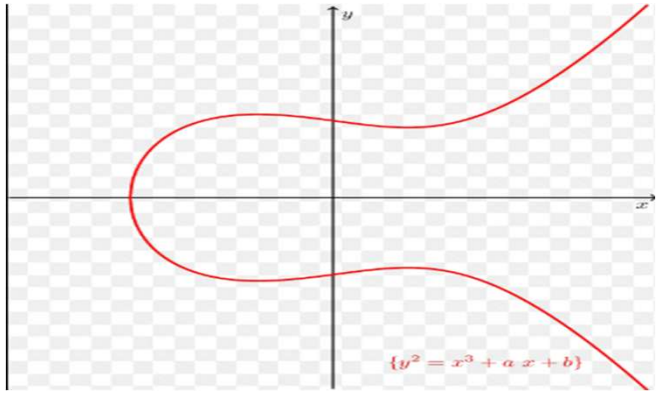
Come funziona?

Cos'è una curva ellittica

- Una curva ellittica è descritta da un'equazione di questo tipo:

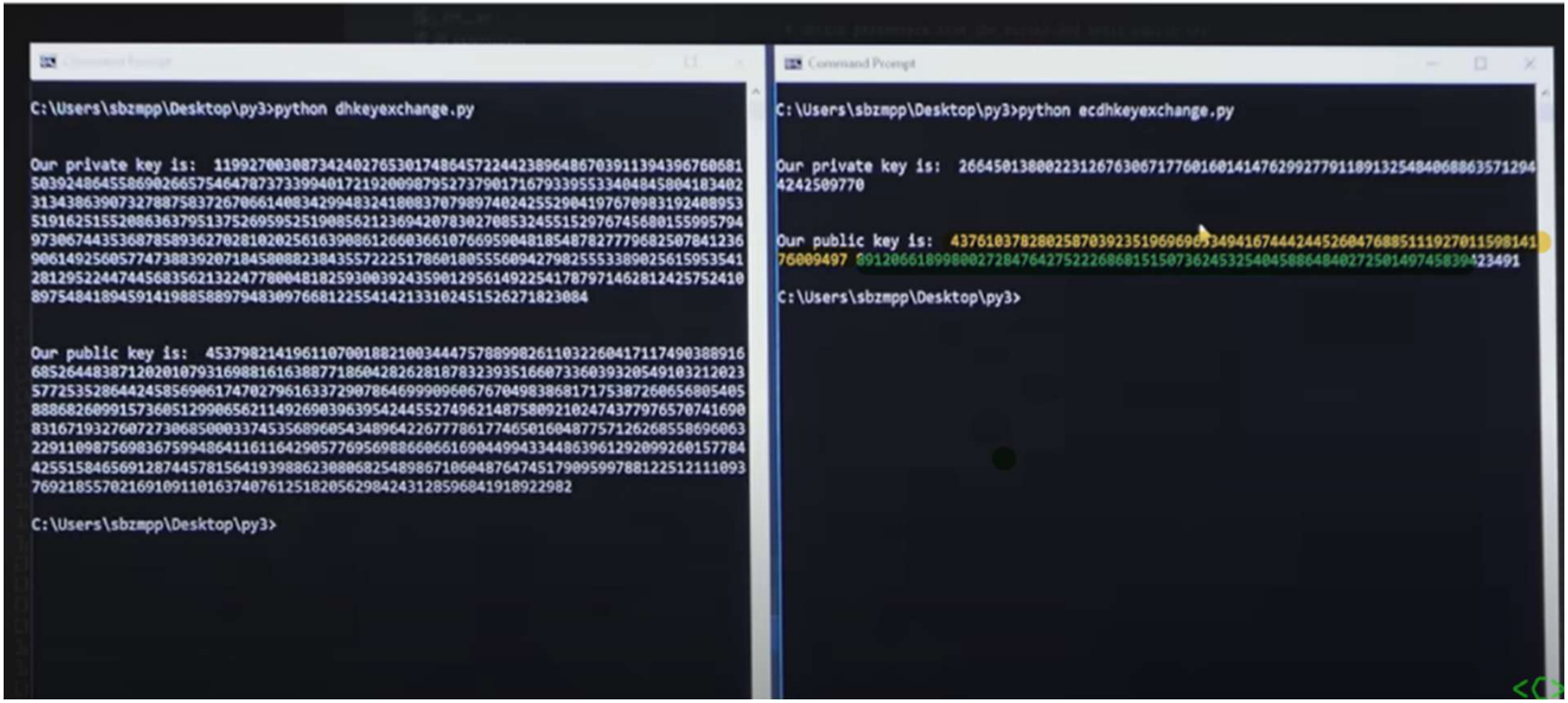
$$y^2 = x^3 + ax + b$$

Su rappresentazione grafica, appare come una curva "morbida" e continua.



In crittografia non lavoriamo con tutti i numeri reali, ma con un insieme finito di valori (**modulo p**). Questo trasforma la curva in una serie di punti discreti (come un reticolo). Su questa curva si troveranno delle coordinate che identificherà la chiave pubblica.

Coordinate di ECC



Applicazione pratica delle normative

Data at rest

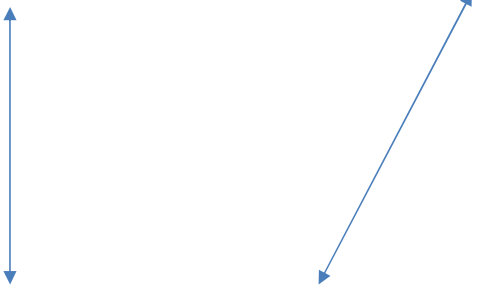
- Crittografia di chiavi simmetriche per protezione database e archivi
- Esempi pratici: dati sanitari su server, documenti legali archiviati.

Dati in transit

- Scambio di chiavi ECDH (Elliptic Curve Diffie-Hellman) per stabilire canali sicuri
- Applicazione GDPR: connessioni HTTPS/TLS ottimizzate, IoT telemetry protetta

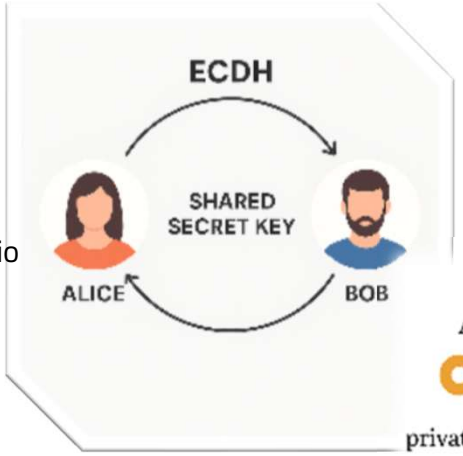
Dati in use

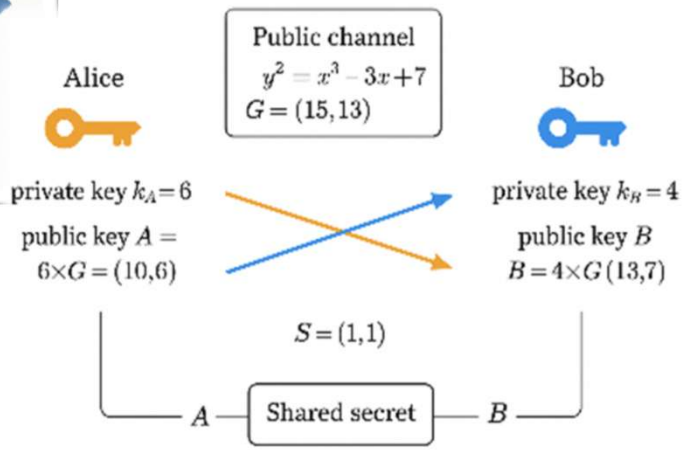
- Firme digitali ECDSA per autenticazione e integrità dei dati in elaborazione
- Funzione specifica: validazione di transazioni blockchain e sessioni SSH sicure



ECC offre sicurezza comparabile a RSA con chiavi più piccole, riducendo l'overhead e supportando i requisiti di conformità all'Articolo 32 GDPR

Esempio di Alice e Bob





Passo 1: Alice e Bob si mettono d'accordo pubblicamente su:

- Una curva ellittica definita da parametri standard (per esempio secp256r1)
- Un punto generatore G sulla curva (noto a entrambi)
- Il campo finito su cui lavorano (modulo p)

Passo 2: Scelta di chiavi private (segrete)

- Alice sceglie un numero segreto casuale k_A (la sua chiave privata)
- Bob sceglie un numero segreto casuale k_B (la sua chiave privata)

Passo 3: Calcolo chiavi pubbliche

- Alice calcola la sua chiave pubblica $K_A = k_a \cdot G$
- Bob calcola la sua chiave pubblica $K_b = k_b \cdot G$

Passo 4: Scambio chiavi pubbliche:

- Alice manda K_A a Bob
- Bob manda K_B ad Alice

Passo 5: Calcolo chiave segreta condivisa:

Alice calcola $S = k_A \cdot K_B$
Bob calcola $S = k_B \cdot K_B$
Per la proprietà del gruppo sulla curva, $k_A \cdot K_B = k_A \cdot (k_B \cdot G) = k_B \cdot (k_A \cdot G) = k_B \cdot K_A$ entrambi ottengono lo stesso punto S.

Passo 6: Uso della chiave segreta:

- Dal punto S, Alice e Bob estraggono un valore numerico (ad esempio la coordinata x) da usare come chiave simmetrica per cifrare le comunicazioni future.

Da cosa nasce la sicurezza?

Un eventuale attaccante, pur conoscendo G, K_A, e K_B, non può risalire ai valori privati k_A o k_B grazie alla difficoltà del problema del logaritmo discreto su curva ellittica.

Prestazioni migliori grazie ad ECC

encrypted secret share with Paillier requires additional highly cost range checks so it take a long time and a large amount of energy in key generation phase and signature generation phase, which is not appropriate for modern communication. In this context, the ECDSA is considered suitable for IoT-devices as it uses small key sizes, which decreases computation time by 65% and 87%, and reduces energy consumption by 77% and 82% respectively, and makes it more efficient and performs faster than other protocols.

Secondo un caso studio pubblicato *sull'International Journal of Computer Networks & Communications (IJCNC) Vol.15, No.4, July 2023*, ECC riduce significativamente sia il consumo energetico (fino all'82%) che il tempo di calcolo (fino all'87%) rispetto a RSA nei sistemi IoT.
Miglioramento importante, per i sistemi hardware computazionalmente meno performanti dei dispositivi IoT.

Lo stesso studio evidenzia che il rapporto di efficienza è impressionante: per ottenere 128 bit di sicurezza, ECC richiede solo 256 bit di chiave contro i 3072 bit necessari per RSA, rappresentando una riduzione del 91% nella dimensione della chiave.

L’effettiva efficienza di ECC

- Standard Governativi e Istituzionali
  - NIST ha standardizzato 15 curve ellittiche per uso governativo americano. Le curve NIST (secp256r1, secp384r1, secp521r1) sono ampiamente adottate.
  - L'affidabilità di ECC è confermata dall'adozione da parte del NIST (National Institute of Standards and Technology), che ha standardizzato 15 curve ellittiche specifiche per l'uso in applicazioni governative e di sicurezza nazionale.

- Adozione nelle Criptovalute
  - Bitcoin utilizza la curva secp256k1, mentre Ethereum implementa ECC per le operazioni crittografiche. Queste sono le due criptovalute più importanti al mondo.

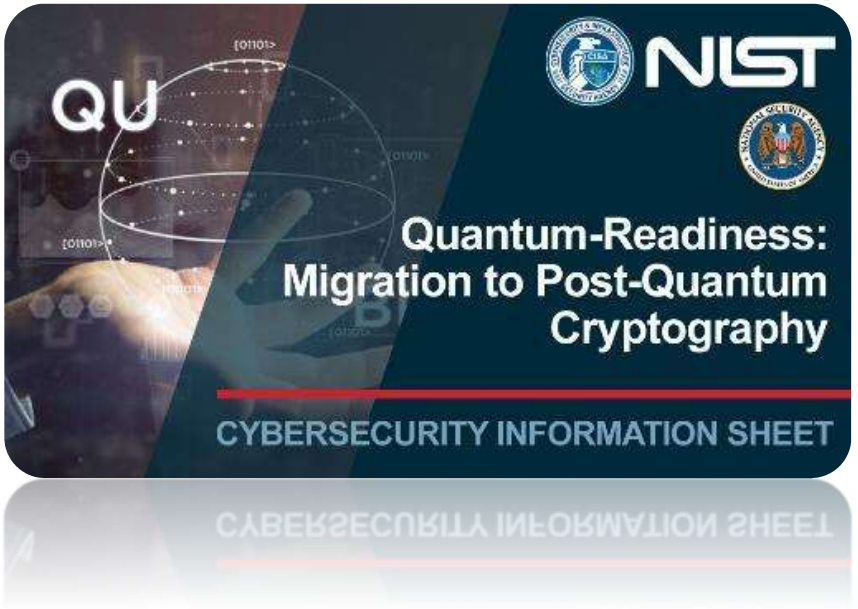
- Resistenza Quantistica Relativa
  - Sebbene nessun algoritmo crittografico attuale sia completamente immune ai computer quantistici, ECC mostra una maggiore resistenza rispetto a RSA contro potenziali attacchi quantistici futuri.

Crittografia Post-Quantum

Come nasce?

- Come già discusso nelle sezioni precedenti, gli algoritmi di crittografia attualmente utilizzati (RSA, Diffie-Hellman, ECC) sono considerati sicuri contro attacchi condotti con i computer "classici". Tuttavia, con l'avanzare della ricerca nel campo del **calcolo quantistico**, questi stessi algoritmi rischiano di essere facilmente compromessi.

- Per affrontare questa minaccia, la comunità scientifica ha sviluppato il concetto di **crittografia post-quantum (Post-Quantum Cryptography, PQC)**, ossia l'insieme di algoritmi progettati per resistere ad attacchi da parte di computer quantistici.



- Il National Institute of Standards and Technology (NIST) definisce la crittografia post-quantistica (PQC) come l'insieme di algoritmi crittografici che utilizzano modelli matematici classici ma resistono agli attacchi sia dei computer quantistici che di quelli classici.

Il NIST però ha avuto un ruolo centrale: nel 2016 ha lanciato un processo ufficiale di **standardizzazione** della crittografia post-quantum, che ha dato visibilità e struttura a tutto il settore.

Shor e Grover: la sfida del quantum computing

Le prime minacce alla crittografia moderna compaiono già tra il 1994 e il 1996, dagli informatici e matematici ai Bell Labs, Peter Shor e Lov Grover, quando vengono proposti i primi algoritmi quantistici in grado di ridurre drasticamente la sicurezza dei sistemi esistenti: l'algoritmo di Shor e l'algoritmo di Grover.

Questi due algoritmi hanno mostrato come il calcolo quantistico possa mettere in crisi la crittografia classica. Shor mina gli algoritmi a chiave pubblica (RSA, Diffie-Hellman, ECC), mentre Grover riduce la sicurezza di algoritmi simmetrici e funzioni hash.

Perché sono una minaccia?

Il motivo principale è dato dal concetto di qubit:

Nei computer classici: bit = 0 oppure 1
Nei computer quantistici: qubit = 0 e 1 insieme (sovrapposizione)
Grazie a sovrapposizione ed entanglement, un registro di qubit elabora moltissime combinazioni in parallelo→ Questo "calcolo parallelo naturale" rende algoritmi come Shor e Grover molto più efficienti di quelli classici.

Panoramica degli algoritmi

Algoritmo di Shor (1994)

Target: Crittografia asimmetrica

Complessità: $O(n^3)$ - Polinomiale

Algoritmi vulnerabili:

- RSA
- ECC (Elliptic Curve)
- Diffie-Hellman

Impatto: RSA 2048-bit diventa completamente insicuro

Algoritmo di Grover (1996)

Target: Crittografia simmetrica

Complessità: $O(\sqrt{N})$ - Radice quadrata

Algoritmi vulnerabili:

- AES
- Funzioni hash

Impatto: Riduzione 50% sicurezza (AES-128 → 64-bit)

Algoritmi di Difesa Post-Quantum: Kyber e Dilithium

Abbiamo analizzato gli algoritmi quantistici, nati come 'offesa' per le tecniche crittografiche attuali; dal progresso quantistico sono però emersi anche algoritmi di difesa.

- Kyber – Proteggere lo scambio di chiavi

Kyber è pensato per "impacchettare" e trasmettere in modo sicuro la chiave segreta che due parti useranno poi per cifrare i dati. Con Kyber è possibile farlo anche in un mondo in cui i computer quantistici esistono, perché si basa su problemi matematici difficilissimi da risolvere, anche con un computer quantistico.

- Dilithium – Autenticare messaggi e software

Dilithium serve a firmare digitalmente documenti o software, garantendo che l'autore sia davvero chi dice di essere e che il contenuto non sia stato modificato. È l'equivalente post-quantum di RSA o ECDSA: ti permette di verificare firme sicure anche quando i computer quantistici diventeranno abbastanza potenti da rompere gli schemi attuali.

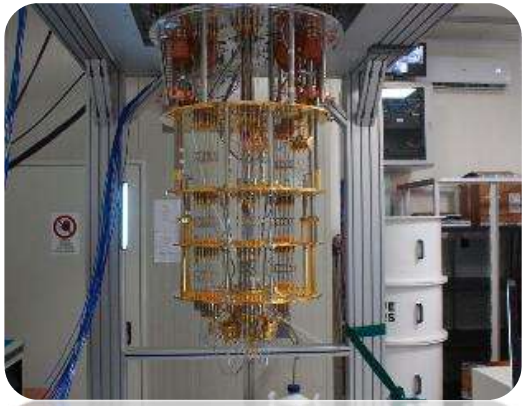
Algoritmi quantistici: minaccia imminente?

Nonostante i timori legati alla potenza dei computer quantistici, la loro minaccia reale alla crittografia moderna non è ancora imminente.

- Ad esempio, per rompere RSA con l'algoritmo di Shor sarebbero necessari circa 10.000 qubit logici.

Considerando l'attuale tecnologia, questo corrisponderebbe a decine o centinaia di migliaia di qubit fisici.

In confronto, i computer quantistici attualmente presenti in Italia sono molto più piccoli.



In Italia il computer quantistico del Politecnico di Torino dispone di 4 qubit fisici, mentre quello dell'Università Federico II di Napoli ne ha 24.

Entro il 2033, IBM ha annunciato l'obiettivo di sviluppare un supercomputer quantistico da 100.000 qubit, collaborando con istituzioni come l'Università di Tokyo e l'Università di Chicago. Questo progetto ambizioso sottolinea la rapida evoluzione della tecnologia quantistica.

Grazie per l'attenzione