ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

CAMPUS DI CESENA
DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA
Corso di Laurea in Ingegneria e Scienze Informatiche

SICUREZZA ED INTEGRITÀ IN BITCOIN: UN'ANALISI DEI PILASTRI CRITTOGRAFICI

Elaborato in
CRITTOGRAFIA

Relatore
Prof. LUCIANO MARGARA

Presentata da
BARTOLINI RICCARDO

Anno Accademico 2024 – 2025

Indice

1 Introduzione

La crittografia rappresenta una delle discipline centrali della sicurezza informatica moderna. Originariamente concepita come arte di nascondere messaggi in contesti militari e diplomatici, si è evoluta in una scienza rigorosa fondata su basi matematiche solide. I primi cifrari classici — come il cifrario di Cesare o quello di Vigenère — oggi hanno soprattutto un valore storico e didattico, ma hanno aperto la strada a un percorso di crescente sofisticazione, culminato con i contributi di Claude Shannon, che ha formalizzato concetti come l'entropia informativa e i requisiti di sicurezza perfetta. In questa cornice, la crittografia contemporanea si articola attorno a quattro obiettivi fondamentali, tra loro interdipendenti e alla base della sicurezza digitale: *riservatezza*, che limita l'accesso ai dati agli attori autorizzati; *integrità*, che rende rilevabili le modifiche non autorizzate; *autenticazione*, che attesta l'identità delle parti; e *non ripudio*, che rende innegabile l'origine di un messaggio o di una transazione. Queste garanzie permeano la vita quotidiana, dall'e-commerce alle comunicazioni cifrate in tempo reale, fino ai pagamenti elettronici.[3] [6]

Su queste fondamenta si innestano strumenti moderni che combinano matematica, teoria della complessità e ingegneria dei sistemi. Le funzioni di hash crittografiche (ad es. SHA-256) realizzano digest a lunghezza fissa con proprietà di preimage e collisione *computazionalmente difficili* da violare; gli schemi di firma digitale (in particolare ECDSA, e più recentemente Schnorr) consentono autenticazione e non ripudio a partire da chiavi pubbliche su curve ellittiche; strutture dati come i *Merkle tree* abilitano verifiche efficienti su grandi insiemi di transazioni riducendo l'onere per i nodi leggeri. La sicurezza di tali primitive poggia sull'asimmetria fra problemi "facili" da calcolare e "difficili" da invertire, cardine che rende praticabile la separazione fra chiave pubblica e privata e sostiene l'affidabilità degli schemi di firma.

Bitcoin nasce precisamente all'incrocio fra queste idee: combina funzioni hash, firme a chiave pubblica e una rete peer-to-peer per coordinare attori non fidati senza autorità centrale. La *Proof-of-Work* (PoW) sincronizza la produzione dei blocchi e rende costosa la riscrittura della storia; le firme digitali vincolano la spesa degli output alle chiavi dei legittimi proprietari; i Merkle tree comprimono l'insieme delle transazioni in ogni blocco, consentendo verifiche efficienti. Nel tempo, gli avanzamenti crittografici hanno guidato evoluzioni concrete del protocollo: *Segregated Witness* (SegWit) ha separato i dati di firma risolvendo la malleabilità e ottimizzando la capacità effettiva; *Taproot*, con l'adozione delle firme di Schnorr e la costruzione *Merkleized* di condizioni di spesa, ha portato miglioramenti in efficienza e privacy senza alterare i principi fondanti del sistema.

Nonostante la robustezza dell'impianto, emergono sfide sistemiche. La **scalabilità** è limitata dalla dimensione dei blocchi e dai tempi medi di conferma: il throughput on-chain non può crescere indefinitamente senza intaccare la decentralizzazione (più dati, nodi più costosi). Per rispondere, la comunità ha esplorato strategie complementari. Sul *Layer 1* l'ottimizzazione dello spazio (ad es. con SegWit) ha mitigato alcune pressioni; sul *Layer 2* sono nati protocolli come il *Lightning Network*, che spostano gran parte delle transazioni *off-chain* attraverso canali di pagamento bidirezionali, con instradamento multi-hop e finalità crittograficamente garantita. In parallelo, l'uso di *sidechain* consente di sperimentare funzionalità e regole diverse mantenendo il *Layer 1* come strato di regolamento sicuro. Queste scelte ingegneristiche non sono mai solo tecniche: toccano l'equilibrio fra sicurezza, decentralizzazione ed efficienza, e si intrecciano con questioni di governance.

La **privacy** è un altro tema cruciale. La pseudonimia nativa di Bitcoin non equivale ad anonimato: tecniche di analisi della rete e del grafo di transazioni possono indebolire la riservatezza degli utenti. Ne è seguita un'intensa ricerca su protocolli e pratiche: schemi di *mixing* come *CoinJoin*/*PayJoin* e costruzioni operative (ad es. Stonewall/Stonewallx2) aumentano la non-distinguibilità degli schemi di spesa; *Dandelion++* agisce a livello di rete per offuscare la sorgente di propagazione; l'uso consapevole degli output (evitando il riuso di indirizzi, curando il resto, consolidando UTXO a fee basse) riduce le correlazioni. Sul fronte teorico, le *Zero-Knowledge Proofs* (ZKP) promettono di dimostrare proprietà sulle transazioni senza rivelarne i dettagli, aprendo percorsi verso una maggiore riservatezza compatibile con la trasparenza di un registro pubblico.

Infine, la **sfida quantistica** pone questioni prospettiche; gli schemi su curve ellittiche oggi impiegati sono ritenuti sicuri nel modello classico, ma l'eventuale disponibilità di calcolatori quantistici su larga scala richiederebbe la migrazione a primitive *post-quantum*. Bitcoin, per la sua diffusione e per i vincoli di compatibilità e decentralizzazione, rappresenta un banco di prova esigente per l'adozione di standard PQC: bilanciare sicurezza, efficienza e implementabilità sarà determinante per preservare il valore di lungo periodo.

Obiettivo e impostazione del lavoro: Questa tesi esamina i pilastri crittografici che sostengono Bitcoin, mettendone in luce solidità, vulnerabilità e traiettorie di evoluzione. L'approccio è al tempo stesso descrittivo — per illustrare componenti e meccanismi fondamentali — e critico, per evidenziare limiti, vettori d'attacco e possibili soluzioni. In particolare, si (i) ricapitolano le basi della crittografia (simmetrica, asimmetrica, hash) e i requisiti di sicurezza; (ii) si analizzano gli elementi crittografici di Bitcoin (hashing, Merkle tree, firme ECDSA/Schnorr) e la loro integrazione nel protocollo; (iii) si discutono modelli e pratiche di gestione delle chiavi e dei wallet; (iv) si studiano le principali superfici d'attacco e i meccanismi di difesa, con attenzione al ruolo della PoW e dell'aggiustamento di difficoltà; (v) si affrontano le tecniche per la privacy on-chain e a livello di rete e le prospettive delle ZKP; (vi) si valutano le strategie di scalabilità on-chain e off-chain (Lightning, sidechain) e le implicazioni di governance (hard/soft fork); (vii) si considerano gli scenari post-quantum e le ricadute per la resilienza del sistema. L'auspicio è offrire una visione coerente e completa del ruolo della crittografia in Bitcoin e delle direzioni di ricerca necessarie a garantirne robustezza e sostenibilità nel tempo.

2 Crittografia di base

2.1 Definizioni ed obiettivi della crittografia

La crittografia è la scienza e l'arte di cifrare e decifrare le informazioni, trasformando i dati in un formato illeggibile senza l'ausilio di una chiave specifica. Essa si configura come un sottocampo della crittologia, una disciplina più ampia che include anche la crittoanalisi, focalizzata sulla decifrazione dei messaggi cifrati, e la steganografia, che si occupa della dissimulazione di messaggi segreti. Gli obiettivi primari della crittografia sono molteplici e interconnessi, formando una gerarchia di necessità per qualsiasi comunicazione o transazione digitale sicura.
La loro integrazione simultanea è cruciale per l'efficacia di un sistema crittografico:

- **Riservatezza (Confidentiality:)** L'obiettivo principale è rendere le informazioni disponibili esclusivamente agli utenti autorizzati, proteggendole da accessi non desiderati.
- **Integrità (Integrity):** Questo principio garantisce che le informazioni non siano state manipolate o alterate durante la trasmissione o l'archiviazione. Anche la minima modifica ai dati deve essere rilevabile.
- **Autenticazione (Authentication):** La crittografia consente di confermare l'autenticità delle informazioni o l'identità di un utente, assicurando che i dati provengano dalla fonte dichiarata.
- **Non Ripudio (Non-Repudiation):** Questo obiettivo impedisce a un utente di negare impegni o azioni precedentemente intraprese, fornendo una prova inconfutabile dell'origine di un'azione o di un messaggio.

Questi quattro obiettivi non sono concetti isolati, ma costituiscono i pilastri interdipendenti della sicurezza digitale. La riservatezza è un punto di partenza essenziale, ma senza l'integrità, un messaggio potrebbe essere modificato senza preavviso. L'autenticazione è necessaria per stabilire la provenienza, e il non ripudio fornisce una prova irrefutabile dell'origine. La solidità di sistemi complessi come Bitcoin deriva proprio dalla capacità di integrare algoritmi e protocolli che soddisfino contemporaneamente tutti questi aspetti. Questo spiega la ragione per cui Bitcoin impiega una combinazione di funzioni hash per l'integrità, firme digitali per l'autenticazione e il non ripudio, e crittografia asimmetrica per la gestione delle chiavi che abilita sia la riservatezza che l'autenticazione delle transazioni.[1] [2]

2.2 Tipologie di crittografia

La crittografia moderna si articola principalmente in tre categorie: crittografia simmetrica, crittografia asimmetrica e funzioni hash crittografiche, ognuna con principi e applicazioni distinti.

2.2.1 Crittografia simmetrica (a chiave segreta)

La crittografia simmetrica, nota anche come crittografia a chiave segreta, impiega una singola chiave condivisa per entrambe le operazioni di cifratura e decifratura dei dati. Questo metodo richiede che mittente e destinatario abbiano accesso alla stessa chiave, il

che implica la necessità di un canale sicuro per la sua condivisione iniziale. Dal punto di vista computazionale, la crittografia simmetrica è meno onerosa e significativamente più efficiente per la gestione di grandi volumi di dati rispetto ai metodi asimmetrici. Questa efficienza computazionale non è un mero dettaglio tecnico, ma un fattore determinante per la sua adozione pratica su larga scala. Sebbene la sua sicurezza intrinseca possa essere percepita come inferiore a causa della necessità di condividere la chiave, la sua velocità la rende indispensabile per applicazioni che elaborano volumi elevati di dati. Ciò porta all'adozione di approcci ibridi, dove la crittografia asimmetrica viene utilizzata per lo scambio sicuro di chiavi simmetriche, combinando così i punti di forza di entrambe le tipologie. Tra gli algoritmi più comuni e riconosciuti in questa categoria si annoverano:

- **Advanced Encryption Standard (AES):** Generalmente considerato il migliore e più ampiamente adottato a livello globale, incluso dal governo degli Stati Uniti. Offre una robusta sicurezza con lunghezze di chiave di 128, 192 o 256 bit.
- **Data Encryption Standard (DES) e Triple DES (3DES):** Predecessori di AES, ancora in uso in alcuni contesti, ma meno sicuri rispetto ad AES.
- **Twofish e Blowfish:** Altri algoritmi simmetrici noti per la loro robustezza e velocità.

I cifrari simmetrici si distinguono in due sottocategorie principali:

- **Cifrari a blocchi:** Come AES, crittografano i dati in blocchi di dimensioni fisse (es. 128 bit).
- **Cifrari a flusso:** Come RC4, crittografano i dati un bit o un byte alla volta, rendendoli adatti per l'elaborazione di dati in tempo reale.

I casi d'uso della crittografia simmetrica sono ampi e fondamentali per le moderne pratiche di sicurezza dei dati. La sua efficienza e semplicità la rendono la scelta preferita per:

- **Sicurezza di grandi quantità di dati:** Protezione di informazioni sensibili da attacchi informatici.
- **Comunicazioni e navigazione web sicure:** Utilizzata in protocolli come Transport Layer Security (TLS) per salvaguardare l'integrità e la riservatezza dei dati trasmessi su internet (e-mail, messaggistica istantanea, HTTPS).
- **Sicurezza del cloud e crittografia di database:** Protezione di dati sensibili sia on-premise che in ambienti cloud.
- **Integrità dei dati:** Generazione di Message Authentication Code (MAC) per confermare che i dati non siano stati alterati.
- **Crittografia di file, cartelle e dischi:** Protezione di dati archiviati su sistemi locali e supporti rimovibili.
- **Conformità normativa:** Aiuto alle organizzazioni nel rispettare i requisiti normativi per la protezione dei dati sensibili.
- Cifratura: $\mathbf{C = E\_k(P)}$ (dove C è il testo cifrato, E è la funzione di cifratura, k è la chiave segreta, P è il testo in chiaro).
- Decifratura: $\mathbf{P = D\_k(C)}$ (dove D è la funzione di decifratura).

Figura 1: Questa immagine illustra il funzionamento di base della crittografia simmetrica, mostrando come un messaggio in chiaro venga cifrato e decifrato utilizzando la stessa chiave.

[5]

2.2.2 Crittografia asimmetrica (a chiave pubblica)

A differenza della crittografia simmetrica, la crittografia asimmetrica, o a chiave pubblica, impiega una coppia di chiavi matematicamente correlate: una chiave pubblica, che può essere liberamente distribuita, e una chiave privata, che deve rimanere segreta. La sicurezza di questo sistema si fonda sulla difficoltà computazionale di derivare la chiave privata dalla chiave pubblica. Un messaggio cifrato con una delle due chiavi può essere decifrato solo con l'altra. [4]
La dualità chiave pubblica/privata è il cuore della crittografia asimmetrica e rappresenta il fondamento della fiducia decentralizzata. La proprietà che la chiave privata non è ricavabile dalla chiave pubblica non solo abilita la riservatezza delle comunicazioni, ma, in modo cruciale, consente la "firma digitale". La firma digitale, a sua volta, è il meccanismo che permette l'autenticazione e il non ripudio senza la necessità di un'autorità centrale che certifichi l'identità. Questo è il meccanismo abilitante per sistemi decentralizzati come Bitcoin, dove la capacità di firmare digitalmente transazioni con una chiave privata, senza che questa possa essere falsificata o derivata dalla chiave pubblica, è l'unico modo affidabile per provare la proprietà e autorizzare la spesa di fondi in un ambiente senza fiducia. La crittografia asimmetrica, quindi, non è solo uno strumento di sicurezza, ma un paradigma che rende possibile la decentralizzazione.[9]
Gli usi principali della crittografia asimmetrica sono:

- **Crittografia a Chiave Pubblica:** Il mittente cifra un messaggio utilizzando la chiave pubblica del destinatario. Solo il possessore della chiave privata corrispondente può decifrare il messaggio, garantendone la riservatezza.
- **Firma Digitale:** Il mittente firma un messaggio utilizzando la propria chiave privata. Chiunque disponga della chiave pubblica del mittente può verificare la firma, provando l'autenticità del mittente e l'integrità del messaggio (ossia che non è stato alterato).

[11]

Algoritmi comuni in questa categoria includono:

- **RSA (Rivest-Shamir-Adleman):** Ampiamente utilizzato, la sua sicurezza si basa sulla difficoltà computazionale di fattorizzare numeri interi molto grandi.
- **Elliptic Curve Cryptography (ECC):** Offre un'elevata sicurezza con chiavi di dimensioni inferiori rispetto a RSA, basandosi sulla difficoltà del problema del logaritmo discreto su curve ellittiche.
- **Digital Signature Algorithm (DSA):** Un algoritmo specifico per la generazione di firme digitali.
- **Diffie-Hellman (DH):** Un algoritmo crittografico a chiave pubblica creato specificamente per aiutare le parti a concordare su una chiave simmetrica in assenza di un canale sicuro.

La crittografia asimmetrica è impiegata in numerosi contesti, inclusi:

- Comunicazione sicura su reti non affidabili.
- Protocolli Internet come TLS/HTTPS (per lo scambio di chiavi simmetriche e l'autenticazione del server), S/MIME, PGP, GPG, SSH.
- Reti private virtuali (VPN) e sistemi di posta elettronica sicura.
- Tecnologia blockchain, dove è fondamentale per la gestione delle chiavi e la firma delle transazioni.

[10]

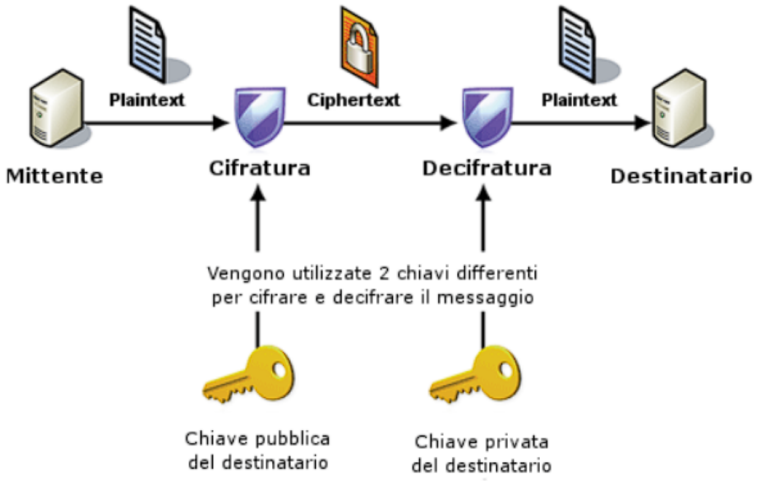

Figura 2: Questa immagine rappresenta il processo di cifratura asimmetrica, dove una chiave pubblica viene usata per cifrare e una chiave privata per decifrare.

- Cifratura: $\mathbf{C = E\_pub(P)}$ (dove pub è la chiave pubblica del destinatario).
- Decifratura: $\mathbf{P = D\_priv(C)}$ (dove priv è la chiave privata del destinatario).

2.2.3 Funzioni Hash Crittografiche

Le funzioni hash crittografiche sono strumenti essenziali che trasformano qualsiasi quantità di dati in una stringa di bit a lunghezza fissa, nota come valore di hash o "impronta digitale". Queste funzioni sono progettate per produrre sempre lo stesso valore di hash per un dato input.

- **h = H(M)** (dove h è l'hash, H è la funzione hash, M è il messaggio/dato).Non è possibile ricavare M da h(onewayness) ed è estremamente difficile trovare M' tale che H(M) = H(M') con M != M'(resistenza alle collisioni).

Le proprietà chiave delle funzioni hash sono fondamentali per la loro applicazione nella sicurezza digitale:

- **Irreversibilità (One-wayness/Pre-image Resistance):** È computazionalmente impraticabile ricostruire l'input originale a partire dal valore di hash. Questa proprietà impedisce, ad esempio, di recuperare una password dal suo hash memorizzato.
- **Resistenza alle Collisioni (Collision Resistance):** È computazionalmente impossibile trovare due input diversi che producano lo stesso valore di hash. Questa caratteristica è cruciale per l'integrità dei dati, poiché impedisce la creazione di documenti falsi con la stessa impronta digitale di un originale.
- **Effetto Valanga (Avalanche Effect):** Anche la minima modifica all'input originale genera un valore di hash completamente diverso e imprevedibile. Questo rende qualsiasi manomissione dei dati immediatamente rilevabile.
- **Deterministico:** Per un dato input, la funzione hash produrrà sempre lo stesso output, indipendentemente dal numero di volte che viene eseguita.

[21] Le funzioni hash sono descritte come "impronte digitali per i dati". Le loro proprietà chiave, come l'irreversibilità, la resistenza alle collisioni e l'effetto valanga, non sono solo caratteristiche matematiche, ma abilitano direttamente la fiducia in sistemi distribuiti. Senza l'effetto valanga, piccole modifiche ai dati non sarebbero rilevabili; senza resistenza alla pre-immagine, le password hashate potrebbero essere invertite; senza resistenza alle collisioni, si potrebbero creare documenti falsi con la stessa impronta. La robustezza delle funzioni hash è un prerequisito per la sicurezza di molte applicazioni crittografiche, inclusa la blockchain di Bitcoin. Sono il "collante" che garantisce che i dati non siano stati manomessi e che le transazioni siano autentiche. La loro natura unidirezionale è ciò che rende il mining di Bitcoin un processo computazionalmente costoso e verificabile, e il Merkle Tree una struttura efficiente per la verifica dell'integrità.
Algoritmi comuni includono la famiglia Secure Hash Algorithm (SHA), sviluppata dalla National Security Agency (NSA) e dal National Institute of Standards and Technology (NIST) degli Stati Uniti. Le varianti più note sono SHA-256, SHA-224, SHA-384 e SHA-512. SHA-256 è particolarmente diffuso e utilizzato in contesti come la sicurezza blockchain, l'hashing delle password e le firme digitali. Altri algoritmi storici includono MD5 e SHA-1, sebbene siano stati identificati con vulnerabilità significative. RIPEMD-160 è un altro algoritmo hash crittografico utilizzato in Bitcoin.
I casi d'uso delle funzioni hash sono molteplici:

- **Integrità dei dati:** Verificano che i dati non siano stati alterati dopo la trasmissione o l'archiviazione.
- **Archiviazione di password:** Le password degli utenti vengono convertite in valori hash prima di essere memorizzate sui server, aumentando la sicurezza.
- **Firme digitali:** Il messaggio viene sottoposto a hashing prima di essere firmato con la chiave privata, garantendo l'integrità del messaggio firmato.
- **Protocolli di autenticazione:** Utilizzate per verificare l'identità degli utenti.

2.2.4 Funzionamento interno di SHA-256

L'algoritmo SHA-256 elabora i messaggi in blocchi da 512 bit, producendo un digest di 256 bit. Ogni blocco viene diviso in 16 parole da 32 bit $(M_0, M_1, \ldots, M_{15})$, poi estese fino a 64 parole tramite la funzione: [7]

$$W_t = \begin{cases} M_t & 0 \leq t \leq 15 \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16} & 16 \leq t \leq 63 \end{cases}$$

dove:

$$\sigma_0(x) = (x \text{ rotr } 7) \oplus (x \text{ rotr } 18) \oplus (x \gg 3)$$

$$\sigma_1(x) = (x \text{ rotr } 17) \oplus (x \text{ rotr } 19) \oplus (x \gg 10)$$

Ogni round $(t = 0, \ldots, 63)$ aggiorna i registri a, b, c, d, e, f, g, h secondo:

$$T_1 = h + \Sigma_1(e) + Ch(e, f, g) + K_t + W_t$$

$$T_2 = \Sigma_0(a) + Maj(a, b, c)$$

$$h = g,\ g = f,\ f = e,\ e = d + T_1,\ d = c,\ c = b,\ b = a,\ a = T_1 + T_2$$

Algoritmo	Output (bit)	Sicurezza stimata	Note
SHA-256	256	128 bit di collisione	Usato in Bitcoin
SHA-3	256	128 bit di collisione	Resistente ad attacchi di estensione
RIPEMD-160	160	80 bit di collisione	Usato per generare indirizzi Bitcoin

Tabella 1: Confronto tra algoritmi di hashing

2.3 Crittografia post-quantistica e quantistica

L'avanzamento della tecnologia quantistica ha introdotto nuove considerazioni nel campo della crittografia, ponendo interrogativi sulla sicurezza a lungo termine degli algoritmi attuali.

2.3.1 La minaccia dei computer quantistici

I computer quantistici, sebbene siano ancora in una fase di sviluppo e non siano attualmente in grado di risolvere i problemi matematici alla base degli algoritmi a chiave pubblica in uso, rappresentano una potenziale minaccia significativa entro qualche decennio. Algoritmi come RSA, la cui sicurezza si basa sulla difficoltà di fattorizzazione di grandi numeri, e quelli basati su curve ellittiche (come ECC, ampiamente utilizzato in Bitcoin), potrebbero diventare vulnerabili agli attacchi quantistici. Questo scenario impone una "preparazione" e una "transizione" verso nuove soluzioni crittografiche.

2.3.2 Crittografia post-quantistica (PQC)

Per preservare l'utilizzo della crittografia asimmetrica di fronte alla minaccia quantistica, la comunità scientifica ha avviato la ricerca di alternative resistenti agli attacchi quantistici, dando origine al campo della crittografia post-quantistica (PQC).
Il National Institute of Standards and Technology (NIST) degli Stati Uniti è in prima linea in questo sforzo di standardizzazione e ha rilasciato i primi standard di crittografia post-quantistica nell'agosto 2024 :

- **ML-KEM (CRYSTALS-Kyber):** Destinato a diventare lo standard primario per la crittografia generale e lo scambio di chiavi. I suoi vantaggi includono chiavi di crittografia relativamente piccole e una buona velocità operativa.
- **ML-DSA (CRYSTALS-Dilithium):** Designato come standard primario per la protezione delle firme digitali.
- **SLH-DSA (Sphincs+):** Uno standard alternativo per le firme digitali, basato su un approccio matematico diverso (basato su hash) e inteso come metodo di backup nel caso in cui ML-DSA dovesse rivelarsi vulnerabile

[12]

Tabella 2: Algoritmi di Crittografia Post-Quantistica

Algoritmo	**Standard FIPS**	**Scopo**	**Base Matematica**	**Vantaggi Principali**
ML-KEM(ex Kyber)	203	Scambio di chiavi (crittografia)	Crittografia basata su reticoli (Lattice)	Chiavi piccole e alta velocità operativa.
ML-DSA(ex Dilithium)	204	Firme digitali	Crittografia basata su reticoli (Lattice)	Efficienza e dimensioni ridotte delle firme.
SLH-DSA(ex Sphinxs+)	205	Firme digitali (alternativa)	Basato su funzioni hash (Hash-Based)	Non richiede un setup fidato, è un backup.

La sicurezza degli algoritmi post-quantum si basa su problemi matematici alternativi a quelli tradizionali, come lo Shortest Vector Problem (SVP) nei reticoli.
La discussione sulla crittografia post-quantistica rivela che la sicurezza crittografica non è un traguardo statico, ma un processo di adattamento continuo. La minaccia dei computer quantistici, sebbene non imminente per gli attuali sistemi, impone una preparazione e una transizione. Questo implica che anche algoritmi robusti oggi, come quelli usati in Bitcoin, hanno una "data di scadenza" teorica. Per Bitcoin, l'adozione di algoritmi resistenti ai computer quantistici sarà una sfida significativa, che potrebbe richiedere potenziali aggiornamenti al protocollo di base, come hard fork o soft fork, a seconda dell'implementazione. Questo aspetto sottolinea che l'immutabilità della blockchain si riferisce ai dati registrati, ma non necessariamente al codice sottostante, che deve evolvere per mantenere la sicurezza a lungo termine di fronte a nuove minacce computazionali.[13]

2.3.3 Crittografia quantistica (QKD - Quantum Key Distribution)

Prima dell'avvento della PQC, la meccanica quantistica era già stata impiegata nella definizione di nuovi algoritmi crittografici, dando vita alla cosiddetta crittografia quantistica. La Quantum Key Distribution (QKD) è un esempio di questa branca, che mira a distribuire chiavi crittografiche in modo intrinsecamente sicuro sfruttando i principi della meccanica quantistica. Tuttavia, la crittografia quantistica presenta alcune limitazioni pratiche:

- **Sensibilità ai disturbi:** I fotoni, anche in transito, possono essere soggetti a cambiamenti di polarizzazione a causa di fattori esterni, aumentando le possibilità di errore nella trasmissione del messaggio.
- **Costi elevati:** L'installazione delle infrastrutture necessarie per la QKD comporta costi molto elevati.
- **Applicazioni limitate:** Attualmente, le applicazioni della crittografia quantistica sono principalmente limitate alla distribuzione delle chiavi. In molti altri contesti,

come la firma qualificata o la posta certificata, non esiste ancora una soluzione quantistica pratica.

2.4 Complessità computazionale e problema del "facile e difficile"

La crittografia asimmetrica, il cuore di sistemi come Bitcoin, si basa su un principio matematico noto come problema a "senso unico" o a "funzione one-way". Questo principio sfrutta l'esistenza di operazioni matematiche che sono semplici da calcolare in una direzione (il cosiddetto problema "facile") ma estremamente difficili da invertire (il problema "difficile"). Ad esempio, è banale moltiplicare due grandi numeri primi per ottenere un prodotto, ma è computazionalmente proibitivo fattorizzare quel prodotto per ritrovare i numeri primi originali. Allo stesso modo, la sicurezza della crittografia a curva ellittica, utilizzata in Bitcoin, si basa sulla difficoltà di risolvere il problema del logaritmo discreto su curve ellittiche.
La solidità di questi sistemi è ulteriormente sostenuta dalla teoria della complessità computazionale, che categorizza i problemi in base alla difficoltà di risolverli. Le classi di complessità P e NP sono centrali in questo contesto:

- **P** include tutti i problemi che possono essere risolti in tempo polinomiale da un algoritmo deterministico, ovvero in un tempo che cresce in modo gestibile all'aumentare delle dimensioni dell'input.
- **NP** comprende i problemi per cui una soluzione data può essere verificata in tempo polinomiale, anche se il processo per trovare tale soluzione potrebbe essere computazionalmente proibitivo.

Il dibattito irrisoluto tra P e NP è cruciale per la crittografia. La sicurezza computazionale di quasi tutti i moderni algoritmi a chiave pubblica, compresi quelli di Bitcoin, si basa sull'assunto che P != NP, ovvero che i problemi "difficili" non possano essere risolti in modo efficiente. Se un giorno si scoprisse un algoritmo efficiente per risolvere un problema NP in tempo polinomiale (ovvero, se si dimostrasse che P=NP), la sicurezza di questi sistemi sarebbe compromessa. La sicurezza di Bitcoin, quindi, non è assoluta, come nel caso di un cifrario di Vernam, ma è intrinsecamente legata all'assenza di un algoritmo efficiente noto che possa violare gli schemi crittografici. Questa dipendenza evidenzia una vulnerabilità fondamentale e non è dovuta a un errore di progettazione, ma è un presupposto su cui si basa l'intera crittografia computazionale.

2.5 Crittografia avanzata per Bitcoin

Oltre agli algoritmi fondamentali, l'evoluzione della crittografia ha introdotto primitive più sofisticate che aprono nuove possibilità per Bitcoin. Una delle più promettenti è la **Zero-Knowledge Proof (ZKP)**. Una ZKP è un protocollo crittografico in cui una parte, il "prover", può dimostrare a un'altra, il "verifier", che un'affermazione è vera, senza rivelare alcuna informazione aggiuntiva oltre alla veridicità dell'affermazione stessa.
Un sistema ZKP deve soddisfare tre proprietà essenziali:

- **Completezza:** Se l'affermazione è vera, un prover onesto può sempre convincere un verifier onesto.
- **Robustezza (Soundness):** Se l'affermazione è falsa, nessun prover disonesto può convincere un verifier onesto, se non con una probabilità trascurabile.
- **Zero-Knowledge:** Se l'affermazione è vera, il verifier non apprende nulla di nuovo se non il fatto che l'affermazione è vera.

Le ZKP hanno un potenziale rivoluzionario per le applicazioni blockchain, poiché consentono di combinare la trasparenza e la verificabilità, che sono i pilastri della decentralizzazione, con la riservatezza delle informazioni sensibili. Queste primitive possono essere utilizzate per migliorare la privacy delle transazioni, verificare l'identità senza rivelare dati personali e consentire l'esecuzione di contratti intelligenti che nascondono la logica interna.

2.6 Bitcoin: Stato dell'Arte e Sfide

Fin dalla sua nascita, Bitcoin è stato progettato come un sistema "trustless", in cui la fiducia non è riposta in un'autorità centrale, ma nella solidità della matematica e degli incentivi economici. La blockchain immutabile e il meccanismo di Proof-of-Work sono i cardini di questa architettura decentralizzata. Tuttavia, il design originale di Bitcoin, pur garantendo una sicurezza senza pari, ha dovuto affrontare sfide significative legate alla privacy, alla scalabilità e alla sicurezza a lungo termine.
Per superare queste sfide, l'ecosistema Bitcoin ha evoluto un'architettura a più livelli, distinguendo tra il **Layer 1**, la blockchain principale, e i **Layer 2**, protocolli e soluzioni costruite al di sopra di essa. Questo approccio modulare permette alla rete di mantenere intatti i suoi principi fondamentali, delegando al tempo stesso le transazioni più frequenti e le funzionalità avanzate a strati superiori più efficienti e scalabili. Questo modello risponde in modo strategico al cosiddetto "trilemma della scalabilità", secondo cui una blockchain può ottimizzare solo due dei tre aspetti: decentralizzazione, sicurezza e scalabilità. Il percorso di Bitcoin ha mostrato la scelta di privilegiare i primi due, lasciando che le soluzioni di Layer 2 risolvessero la sfida della scalabilità.

3 Elementi crittografici di Bitcoin

Bitcoin, la criptovaluta più riconosciuta, si fonda su un'architettura decentralizzata e sicura resa possibile da un'ingegnosa combinazione di principi crittografici e meccanismi di consenso.

3.1 Bitcoin e la tecnologia Blockchain

La tecnologia blockchain, il cuore di Bitcoin, è un database pubblico e distribuito di transazioni digitali. I singoli record, denominati "blocchi", sono collegati tra loro utilizzando la crittografia, formando una catena sequenziale. Una caratteristica fondamentale di questa struttura è che i blocchi possono solo essere aggiunti al database; una volta registrati, non possono essere modificati né eliminati.[14]
I principi fondanti di Bitcoin e della blockchain che lo supporta sono:

- **Decentralizzazione:** A differenza dei sistemi finanziari tradizionali, Bitcoin non è controllato da un'autorità centrale. La rete è peer-to-peer, e ogni nodo mantiene una copia indipendente del registro distribuito pubblico delle transazioni (la blockchain), eliminando così i single points of failure e rendendo il sistema resiliente agli attacchi centralizzati.
- **Immutabilità:** Le transazioni, una volta registrate in un blocco e aggiunte alla catena, non possono essere modificate retroattivamente. L'integrità dei dati è garantita dalla replicazione massiva del database su ogni nodo della rete.
- **Consenso:** L'accordo sulla validità delle transazioni e dei blocchi avviene attraverso regole predefinite, come il meccanismo di Proof-of-Work. La catena di blocchi più lunga, che rappresenta la maggiore potenza di calcolo investita, è universalmente accettata come la versione corretta della storia delle transazioni.
- **Pseudonimia:** Gli utenti di Bitcoin possono generare indirizzi in modo autonomo, senza la necessità di autorizzazioni da parte di un ente centrale, come avviene per i conti bancari. Le transazioni sono legate a questi codici alfanumerici unici (indirizzi) piuttosto che a nomi reali, offrendo un certo grado di privacy, sebbene non un anonimato completo.

La blockchain di Bitcoin è, in essenza, un "sistema di fiducia algoritmica". La sua sicurezza non deriva da un'entità centrale, ma dalla trasparenza delle regole, dalla replicazione massiva dei dati e dalla difficoltà computazionale di alterare la storia. Questo rappresenta un cambiamento di paradigma rispetto ai sistemi finanziari tradizionali, dove la fiducia è riposta in intermediari.[15][16][17][18]

3.2 Chiavi Pubblica e Privata

Nel contesto di Bitcoin, la sicurezza delle transazioni si basa sull'accoppiamento di chiavi pubblica e privata. La chiave privata è un numero casuale segreto da cui viene derivata la chiave pubblica, e da quest'ultima l'indirizzo Bitcoin.

- **Generazione della Chiave Pubblica dalla Chiave Privata:**
  La chiave pubblica viene generata dalla chiave privata attraverso una moltiplicazione su una curva ellittica, specificamente la curva *secp256k1* per Bitcoin:
  ChiavePubblica=ChiavePrivata×G
  dove G è un punto base predefinito sulla curva ellittica,a moltiplicazione è un'operazione complessa che coinvolge l'addizione ripetuta del punto G su se stesso per un numero di volte pari alla chiave privata.
- **Generazione dell'Indirizzo Bitcoin dalla Chiave Pubblica:**
  L'indirizzo Bitcoin è una rappresentazione abbreviata e controllata della chiave pubblica. Il processo prevede due passaggi di hashing sequenziali:
  - Hash1=SHA256(Chiave Pubblica)
  - IndirizzoBitcoin=RIPEMD160(Hash1)
    A questo hash finale vengono poi aggiunti un checksum e una codifica Base58Check per produrre l'indirizzo nel formato standard che vediamo.

3.2.1 Struttura del blocco

Ogni blocco nella blockchain di Bitcoin ha una capacità di dati massima di 1 MB e raggruppa un certo numero di transazioni. Un blocco è composto da due parti principali: l'header e il body.

- **Header del Blocco:** Contiene metadati cruciali per la gestione e la validazione del blocco:
  - **Versione:** Indica la versione del software del protocollo Bitcoin utilizzata per creare il blocco.
  - **Hash del Blocco Precedente (PrevHash):** Un hash di 256 bit che funge da collegamento crittografico al blocco precedente, garantendo l'integrità e la sequenzialità della catena. Questo collegamento è ciò che rende la blockchain immutabile; modificare un blocco precedente altererebbe il suo hash, invalidando tutti i blocchi successivi.
  - **Merkle Root:** Un hash unico che riassume crittograficamente tutte le transazioni contenute nel body del blocco. La sua importanza è approfondita nella sezione successiva.
  - **Timestamp:** Indica il momento in cui il blocco è stato creato, approssimativamente il timestamp dell'ultima transazione inclusa.
  - **Bits:** Un campo che rappresenta la difficoltà target del mining, un valore che i miner devono superare per trovare un hash valido per il blocco.
  - **Nonce:** Un numero arbitrario che i miner modificano ripetutamente per trovare un hash del blocco che soddisfi la difficoltà target, completando così la Proof-of-Work.
- **Body del Blocco:** Contiene l'elenco delle transazioni che sono state raggruppate e validate per essere incluse in quel blocco.

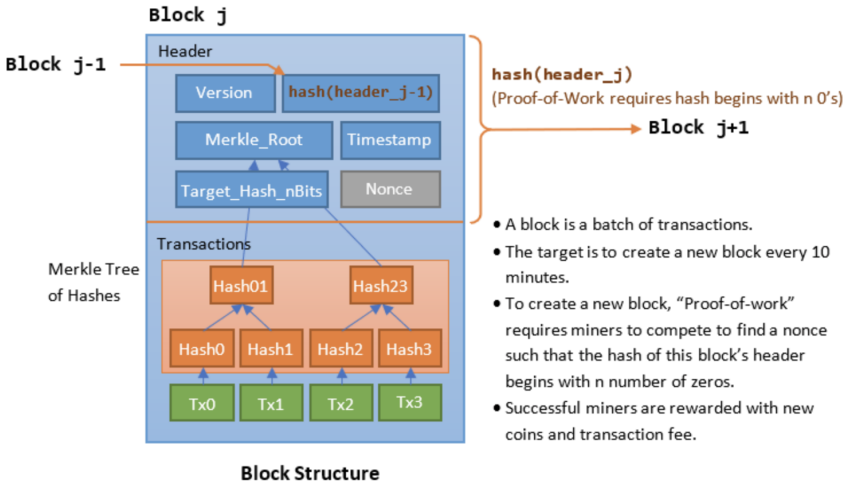

Figura 3: Un diagramma che mostra i componenti principali di un blocco Bitcoin, inclusi l'header, il body e il collegamento al blocco precedente.

3.2.2 Ruoli dei Nodi e dei Miner

I nodi miner svolgono un ruolo vitale nel mantenimento e nella sicurezza della rete Bitcoin. Essi validano le nuove transazioni, le raggruppano in blocchi e competono per trovare una Proof-of-Work valida per il blocco che stanno costruendo. Una volta che un miner trova un blocco valido, lo trasmette al resto della rete, dove viene accettato dagli altri nodi se rispetta le regole del consenso. Per incentivare la loro partecipazione e l'investimento di potenza di calcolo, i miner vengono ricompensati con un "premio del blocco" (nuovi Bitcoin) e le commissioni di transazione pagate dagli utenti. Questo sistema di incentivi è cruciale per la sicurezza e la stabilità della rete decentralizzata.

3.3 Hashing in Bitcoin

L'hashing è un elemento crittografico cardine nel funzionamento di Bitcoin, impiegato per garantire l'integrità dei dati e per il meccanismo di Proof-of-Work.

3.3.1 Uso di SHA-256

Bitcoin utilizza l'algoritmo di hashing SHA-256 (Secure Hash Algorithm 256) per generare hash sicuri e irreversibili. SHA-256 fa parte della famiglia SHA-2, sviluppata dalla NSA e dal NIST, ed è noto per le sue proprietà di resistenza alle collisioni e all'effetto valanga.[20][21]

- **Nel Mining (Proof-of-Work):** Il mining di Bitcoin si basa sulla risoluzione di complessi puzzle matematici. I miner devono trovare un valore nonce che, quando combinato con i dati del blocco (inclusi il PrevHash e il Merkle Root) e sottoposto a hashing, produca un hash del blocco inferiore a un certo target di difficoltà. Bitcoin impiega un "double SHA-256" (SHA-256(SHA-256(data))) per la maggior parte delle operazioni di hashing, inclusi gli header dei blocchi. Questo processo intensivo dal punto di vista computazionale è ciò che valida le transazioni e crea nuovi blocchi, garantendo la sicurezza della rete.

- **Nella Verifica delle Transazioni:** L'hash SHA-256 assicura l'integrità delle transazioni. Anche la minima modifica a qualsiasi dato di input all'interno di una transazione o di un blocco si traduce in un hash completamente diverso, rendendo qualsiasi tentativo di manomissione immediatamente rilevabile.

Ogni blocco nella blockchain di Bitcoin è identificato da un hash unico, calcolato applicando due volte la funzione SHA-256 all'intestazione del blocco. Questo hash funge da impronta digitale del blocco e lo collega in modo immutabile al blocco precedente.
HashBlocco=SHA_256(SHA_256(IntestazioneBlocco))
L'Intestazione Blocco include metadati cruciali come la versione del blocco, l'hash del blocco precedente, la Merkle Root, il timestamp, il 'difficoltà target' e il nonce.

Un aspetto unico dell'implementazione di Bitcoin è l'uso di questo doppio hashing, ovvero l'applicazione della funzione SHA-256 due volte in sequenza per il mining e la creazione degli indirizzi. Sebbene questa pratica possa sembrare una semplice ridondanza per aumentare la sicurezza, la ragione tecnica principale è molto più sofisticata. Il design originale di SHA-256, basato sulla costruzione Merkle-Damgård, è vulnerabile a un tipo di attacco noto come "estensione della lunghezza". Questo attacco consente a un utente malintenzionato, conoscendo l'hash di un messaggio originale, di calcolare l'hash di un messaggio più lungo senza conoscere l'input originale completo. Applicando la funzione hash due volte, si interrompe la catena di dipendenza necessaria per questo tipo di attacco, impedendo all'attaccante di sfruttare la vulnerabilità. Questa scelta di design dimostra una profonda comprensione delle sottigliezze algoritmiche e una sofisticata ingegneria crittografica.

3.3.2 Merkle Tree e Merkle Root

Il Merkle Tree, o albero binario di hash, è un'altra struttura dati crittografica essenziale che organizza tutte le transazioni di un blocco in un'unica "impronta digitale" chiamata Merkle Root. Questa struttura non serve solo a riassumere l'integrità dei dati, ma abilita anche un'importante ottimizzazione per la scalabilità e la verifica dei nodi. [19]
Attraverso le **Merkle Proofs**, un utente può dimostrare che una transazione specifica è inclusa in un blocco senza dover scaricare o rivelare l'intero contenuto del blocco. Per un utente che gestisce un "nodo leggero" (SPV - Simplified Payment Verification), questo riduce drasticamente i requisiti di archiviazione e larghezza di banda, consentendo a un maggior numero di partecipanti di verificare in modo efficiente l'integrità della rete. L'integrazione di Merkle Trees e Merkle Proofs nel protocollo è un esempio di come il design crittografico di Bitcoin supporti direttamente la decentralizzazione, mantenendo il sistema verificabile e accessibile anche a chi non può eseguire un nodo completo.
Processo iterativo Merkle Tree:

- **H_AB = SHA256(H_A || H_B)**
- **H_CD = SHA256(H_C || H_D)**
- **Merkle Root = SHA256(H_AB || H_CD)**
  (dove || indica la concatenazione e H_A, H_B, ecc. sono gli hash delle singole transazioni). Questo dimostra visivamente come l'integrità di tutte le transazioni di un blocco sia riassunta in un singolo hash.[23][24]

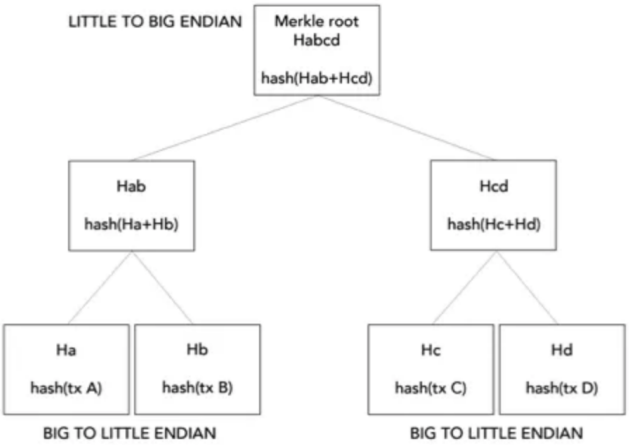

Figura 4: Un diagramma che illustra la struttura di un Merkle Tree, mostrando come gli hash delle transazioni vengano combinati ricorsivamente fino a formare la Merkle Root.

- **Funzionamento:** Le "foglie" alla base dell'albero rappresentano i singoli hash delle transazioni incluse nel blocco. Ogni nodo superiore nell'albero è formato calcolando l'hash della concatenazione dei suoi due nodi figli. Questo processo ricorsivo continua verso l'alto fino a quando non si ottiene un singolo hash in cima all'albero, noto come Merkle Root.
- **Importanza per l'Integrità e la Verifica Efficiente:** Il Merkle Root è un'impronta digitale crittografica che riassume l'intero insieme dei dati sottostanti, ovvero tutte le transazioni di un blocco. Se anche un solo bit di una qualsiasi transazione all'interno del blocco viene modificato, il Merkle Root risultante sarà completamente diverso, segnalando immediatamente la manomissione. Questa proprietà consente ai nodi di verificare l'integrità di un blocco semplicemente controllando il suo Merkle Root, senza dover processare ogni singola transazione.
- **Merkle Proofs:** Le Merkle Proofs sono un'applicazione diretta dei Merkle Tree che consentono a un utente di verificare che una transazione specifica sia inclusa in un blocco senza la necessità di scaricare o rivelare l'intero contenuto del blocco. Per fare ciò, una Merkle Proof include l'hash della transazione in questione e gli hash dei nodi "fratelli" lungo il percorso dalla transazione fino al Merkle Root. Un verificatore, conoscendo il Merkle Root del blocco, può ricalcolare l'hash combinando la transazione con gli hash intermedi forniti e confrontare il risultato finale con il Merkle Root noto. Se i valori coincidono, si ha la certezza che la transazione è inclusa nel blocco. Questa efficienza è particolarmente utile per i "nodi leggeri" (light nodes), che possono così risparmiare banda e spazio di archiviazione.

L'uso di SHA-256 e, in particolare, la struttura del Merkle Tree non sono solo dettagli tecnici, ma elementi fondamentali per la scalabilità e l'integrità di Bitcoin. Il Merkle Root consente di rappresentare milioni di transazioni con un'unica "impronta digitale" di 32 byte. Questo riduce drasticamente la quantità di dati che un nodo deve processare per verificare l'integrità di un blocco. Le Merkle Proofs estendono questo beneficio ai nodi leggeri, che possono verificare l'inclusione di una transazione senza scaricare l'intera blockchain. L'hashing e i Merkle Tree sono soluzioni di scalabilità intrinseche al design

della blockchain di Bitcoin. Permettono al sistema di gestire un volume crescente di transazioni mantenendo la verificabilità per tutti i partecipanti, dai nodi completi ai client leggeri, senza compromettere la sicurezza. Questo è un esempio di come la crittografia sia integrata non solo per la sicurezza, ma anche per l'efficienza operativa.

3.3.3 Esempio di Merkle Proof

Consideriamo 4 transazioni T_1, T_2, T_3, T_4 con hash H_1, H_2, H_3, H_4.

$$H_{12} = SHA256(H_1||H_2), \quad H_{34} = SHA256(H_3||H_4)$$
$$MerkleRoot = SHA256(H_{12}||H_{34})$$

Se vogliamo dimostrare che T_2 appartiene al blocco, basta fornire:

$$Proof = \{H_1, H_{34}\}$$

Il verificatore ricalcola $H_{12} = SHA256(H_1||H_2)$ e poi il Merkle Root.[22]

3.4 Firme digitali in Bitcoin (ECDSA)

Le firme digitali rivestono un ruolo centrale nella sicurezza delle criptovalute, fungendo da meccanismo primario per l'autenticazione e il non ripudio delle transazioni. In Bitcoin, esse sono essenziali per garantire che i fondi possano essere spesi esclusivamente dai loro legittimi proprietari.

3.4.1 Funzionamento di ECDSA

Bitcoin utilizza l'algoritmo ECDSA (Elliptic Curve Digital Signature Algorithm) per generare e verificare le firme digitali. Questo algoritmo si basa sulla crittografia a curva ellittica, che offre un'elevata sicurezza con chiavi di dimensioni relativamente ridotte.[26]

- **Chiave Privata:** La base di una firma digitale è una chiave privata, un numero segreto di 256 bit noto solo alla persona che l'ha generata. Questa chiave è, in sostanza, un numero casuale. In Bitcoin, il possesso della chiave privata corrispondente ai fondi su un indirizzo blockchain conferisce il diritto di spenderli.
- **Chiave Pubblica:** Dalla chiave privata si deriva matematicamente una chiave pubblica. È importante notare che, sebbene la chiave pubblica possa essere calcolata dalla privata, il processo inverso non è computazionalmente fattibile. La chiave pubblica può essere resa nota a chiunque, poiché il suo scopo è unicamente quello di verificare una firma, non di crearla.
- **Generazione della Firma:** Per firmare una transazione (o qualsiasi messaggio), l'algoritmo ECDSA prende un hash del messaggio da firmare (nel caso di Bitcoin, l'hash della transazione) e lo combina con la chiave privata dell'utente. La firma risultante è composta da due numeri, r e s Un elemento di casualità, un numero segreto temporaneo k(nonce), viene introdotto nel processo di firma per garantire che ogni firma generata sia unica, anche se lo stesso messaggio viene firmato più volte con la stessa chiave privata.
  Le formule per la generazione della firma ECDSA sono le seguenti :

  - Generare un numero casuale k (nonce) tra 1 e n-1 (dove n è l'ordine del gruppo della curva).
  - Calcolare (x, y) = k * G, dove G è il punto generatore della curva.
  - Calcolare r = x mod n. Se r=0, generare un nuovo k e ricominciare.
  - Calcolare s = k^-1 * (z + r * d_A_) mod n, dove z è l'hash del messaggio e d_A_ è la chiave privata. Se s=0, generare un nuovo k e ricominciare. La firma è la coppia (r, s).

  [8]

- **Verifica della Firma:** La verifica di una firma digitale avviene utilizzando la chiave pubblica del firmatario, l'hash del messaggio originale e la firma stessa. Un algoritmo matematico specifico permette di determinare se la firma è stata effettivamente prodotta dalla chiave privata corrispondente all'hash e alla chiave pubblica. Se il messaggio viene alterato anche minimamente o se si tenta di verificare la firma con una chiave pubblica diversa da quella del firmatario originale, il processo di verifica fallirà, indicando una manomissione o una non corrispondenza.
  Le formule per la verifica della firma ECDSA sono le seguenti :
  - Verificare che r e s siano entrambi tra 1 e n-1.
  - Calcolare u_1_ = z * s^-1 mod n e u_2_ = r s^-1 mod n.
  - Calcolare (x, y) = u_1_ * G + u_2_ * Q_A_, dove Q_A_ è la chiave pubblica. Assicurarsi che il risultato non sia il punto all'infinito.
  - Se r = x mod n, allora la firma è valida. Altrimenti, o se uno dei controlli fallisce, la firma non è valida.

3.4.2 Riutilizzo del nonce(k)

L'algoritmo di firma digitale a curva ellittica, ECDSA, è il meccanismo che conferisce a Bitcoin il non ripudio e garantisce che solo il legittimo proprietario di una chiave privata possa spendere i fondi. Tuttavia, la sicurezza di ECDSA è estremamente sensibile alla qualità dell'implementazione. Una delle vulnerabilità più critiche riguarda il riutilizzo del nonce, un numero casuale unico utilizzato in ogni singola firma. Se per errore un utente o un'implementazione software riutilizza lo stesso nonce con la stessa chiave privata per firmare due transazioni diverse, un attaccante può facilmente ricavare la chiave privata e rubare i fondi. Questo problema si è manifestato in incidenti reali, come l'hack della PlayStation 3 di Sony e la compromissione di alcuni wallet Bitcoin con generatori di numeri casuali scadenti.

3.4.3 Firme di Schnorr e Taproot

Per migliorare l'efficienza e la sicurezza di questo meccanismo, l'aggiornamento **Taproot**, attivato nel 2021, ha introdotto le Firme di Schnorr come alternativa a ECDSA. Le firme di Schnorr offrono vantaggi significativi, tra cui un minor ingombro on-chain e un'efficienza maggiore. La loro caratteristica più importante è la capacità di **aggregare le firme** , permettendo a più utenti di combinare le loro chiavi pubbliche e le loro firme in un'unica firma valida.

Taproot ha anche introdotto il **MAST (Merkelized Abstract Syntax Tree)**, che consente di comprimere più condizioni di spesa (script) in un'unica Merkle Root. Con MAST, solo la condizione effettiva utilizzata per spendere i fondi viene rivelata on-chain, mantenendo le altre condizioni private. La combinazione di MAST e delle firme di Schnorr ha un impatto profondo sulla privacy. Grazie all'aggregazione delle firme, una transazione che richiede più firmatari (es. un multi-firma o un contratto intelligente) può apparire on-chain come una semplice transazione a firma singola. Questo rende le transazioni complesse indistinguibili da quelle ordinarie, offrendo una "privacy collettiva" che rafforza l'anonimato dell'intera rete, un miglioramento sostanziale rispetto alle tecniche di offuscamento esterne come CoinJoin. La malleabilità delle transazioni è stata risolta con l'implementazione del soft fork SegWit.

Caratteristica	**ECDSA**	**Schnorr**
Dimensione firma	64–72 byte	64 byte fissi
Aggregazione firme	Non supportata	Supportata (multi-firma → firma unica)
Sicurezza formale	Standard NIST, ampiamente testato	Basata su logaritmo discreto, sicurezza dimostrabile
Efficienza	Maggior costo computazionale	Più veloce e semplice
Resistenza alla malleabilità	Vulnerabile	Non vulnerabile

Tabella 3: Confronto tra ECDSA e Schnorr

3.4.4 Esempio numerico ECDSA (curva "giocattolo" su $\mathbb{F}_{17}$)

Consideriamo la curva ellittica $E : y^2 \equiv x^3 + x + 3 \pmod{17}$ sul campo finito $\mathbb{F}_{17}$ e il punto generatore $G = (2, 8)$ di ordine primo $n = 17$ (cioè $17G = \mathcal{O}$, il punto all'infinito). Useremo questo setup per mostrare *firma* e *verifica* ECDSA con numeri piccoli.

Chiavi. Scegliamo una chiave privata $d = 7$ e calcoliamo la chiave pubblica $Q = dG = 7G$. (Con aritmetica sui punti di curva ellittica si ottiene $Q = (11, 6)$.)

Firma. Sia z l'hash del messaggio (ridotto modulo n); fissiamo $z = 13$. Scegliamo un nonce k coprimo con n, ad esempio $k = 5$.

1. Calcola $R = kG = 5G$. Si ottiene $R = (7, 9)$.

$$r \equiv x_R \bmod n \;=\; 7 \bmod 17 \;=\; 7.$$

   Se $r = 0$ si dovrebbe cambiare k; qui va bene.

2. Calcola l'inverso $k^{-1} \bmod n$. Poiché $5 \cdot 7 = 35 \equiv 1 \pmod{17}$, abbiamo $k^{-1} \equiv 7$.

3. Calcola

$$s \equiv k^{-1} \left(z + r\,d\right) \bmod n \;=\; 7 \left(13 + 7 \cdot 7\right) \bmod 17.$$

Poiché $7 \cdot 7 = 49$ e $13 + 49 = 62 \equiv 11 \pmod{17}$, segue

$$s \equiv 7 \cdot 11 \equiv 77 \equiv 9 \pmod{17}.$$

La firma ECDSA sul messaggio (di hash $z = 13$) è quindi la coppia $(r, s) = (7, 9)$.

Verifica. Dati $Q = (11, 6)$, G, $n = 17$, l'hash $z = 13$ e la firma $(r, s) = (7, 9)$:

1. Controlla $1 \leq r, s \leq n - 1$ (vero).
2. Calcola l'inverso $w = s^{-1} \bmod n$. Poiché $9 \cdot 2 = 18 \equiv 1 \pmod{17}$, $w = 2$.
3. Calcola

$$u_1 \equiv z\,w \bmod n \;=\; 13{\cdot}2 \equiv 26 \equiv 9 \pmod{17}, \qquad u_2 \equiv r\,w \bmod n \;=\; 7{\cdot}2 \equiv 14 \pmod{17}.$$

4. Calcola $X = u_1G + u_2Q = 9G + 14Q$. Con l'aritmetica sui punti si ottiene $X = (7, 9)$.
5. La firma è valida se $x_X \bmod n = r$. Qui $x_X = 7$ e $7 \equiv r$, dunque ✓.

Nota. Questo esempio usa una curva minuscola (non sicura) per mostrare i passaggi: in produzione si usano parametri come `secp256k1`, con ordini n di 2^{256}, ma le formule sono esattamente le stesse:

Firma:

$$\begin{cases} r \equiv (kG)_x \bmod n, \\ s \equiv k^{-1}(z + rd) \bmod n \end{cases}$$

Verifica:

$$\begin{cases} w \equiv s^{-1} \bmod n, \\ u_1 \equiv zw \bmod n, \quad u_2 \equiv rw \bmod n, \\ X \equiv u_1G + u_2Q, \\ \text{accetta se } (X_x \bmod n) = r \end{cases}$$

3.4.5 Curva ellittica secp256k1

Satoshi Nakamoto, il creatore di Bitcoin, ha scelto lo standard della curva ellittica secp256k1 per l'implementazione di ECDSA nel protocollo Bitcoin. Questa scelta è stata dettata da diversi fattori: la curva offre un elevato livello di sicurezza, ben testato e riconosciuto, e garantisce un costo computazionale relativamente basso sia per la generazione delle chiavi che per la convalida delle firme. Inoltre, permette la generazione di un numero virtualmente infinito di chiavi pubbliche, supportando la flessibilità necessaria per la gestione degli indirizzi Bitcoin.
ECDSA è il meccanismo che traduce il concetto di "proprietà" in Bitcoin in un formato crittografico verificabile. Il fatto che solo il possessore della chiave privata possa generare una firma valida per spendere i fondi è la garanzia intrinseca che nessuno, tranne il legittimo proprietario, possa spostare i Bitcoin. Questa è la causa diretta della sicurezza delle transazioni in un sistema senza fiducia. La forza di Bitcoin risiede non solo nella blockchain immutabile, ma anche nella robustezza delle firme digitali che controllano l'accesso ai fondi. La gestione sicura della chiave privata diventa, quindi, la responsabilità primaria dell'utente, poiché la sua compromissione annulla la sicurezza fornita da ECDSA, indipendentemente dalla forza dell'algoritmo.[25]

4 Gestione delle chiavi in Bitcoin

La gestione delle chiavi in Bitcoin è un aspetto critico per la sicurezza dei fondi e l'operatività delle transazioni. Essa si basa su una relazione gerarchica tra chiavi private, chiavi pubbliche e indirizzi Bitcoin, supportata da diverse tipologie di portafogli e pratiche di sicurezza.

4.1 Generazione di chiavi e indirizzi Bitcoin

Il processo di generazione di chiavi e indirizzi in Bitcoin segue una logica crittografica precisa:

- Una **chiave privata** è un numero segreto di 256 bit, generato casualmente, che funge da fondamento per l'intero sistema di proprietà dei Bitcoin.
- Da questa chiave privata, attraverso funzioni matematiche unidirezionali (crittografia a curva ellittica), viene derivata una chiave pubblica corrispondente. La chiave pubblica può essere condivisa liberamente, poiché è computazionalmente impossibile risalire alla chiave privata da essa.
- Infine, dalla chiave pubblica viene generato un indirizzo Bitcoin univoco. Questo indirizzo è l'equivalente di un numero di conto bancario, ed è il punto a cui i Bitcoin possono essere inviati.

È fondamentale comprendere che solo il possessore della chiave privata associata a un determinato indirizzo ha la capacità di accedere e spendere i fondi ricevuti su quell'indirizzo.

4.1.1 Tipi di indirizzi Bitcoin

L'ecosistema Bitcoin ha visto l'evoluzione di diversi formati di indirizzi, ciascuno con specifiche caratteristiche e vantaggi, come risposta diretta alle esigenze di rete, in particolare per la scalabilità e la malleabilità delle transazioni. Questa evoluzione dimostra che anche gli aspetti più fondamentali di Bitcoin sono soggetti a miglioramenti per garantire la sostenibilità e l'efficienza a lungo termine della rete. Per gli utenti, ciò implica la necessità di comprendere le differenze tra i tipi di indirizzo per ottimizzare i costi e la compatibilità delle transazioni.

- **Legacy (P2PKH - Pay-to-Pubkey Hash):** Sono i primi formati di indirizzo Bitcoin, riconoscibili perché iniziano con il numero **"1"** (ad esempio, *1A5ZP1E35Qge2zPTuTL5MV7DivFNo*). Sebbene siano ampiamente supportati, sono meno efficienti in termini di spazio e comportano commissioni di transazione più elevate rispetto ai formati più recenti.
- **SegWit (P2WPKH - Pay-to-Witness Public Key Hash):** Questi indirizzi iniziano con **"bc1q"** (ad esempio, *bc1qar0srrr7xfkvy5l643lydnw9re59gtzzwf35f9*). Sono stati introdotti con l'aggiornamento Segregated Witness (SegWit) per ottimizzare la dimensione dei dati delle transazioni, riducendo le commissioni e migliorando l'efficienza e la scalabilità della rete.
- **Compatibilità (P2SH - Pay-to-Script Hash):** Gli indirizzi P2SH iniziano con il numero **"3"** (ad esempio, *3J98T1WPEZ73CNMQVieCRNyiWRNQRHWnly*).

Questo formato consente l'implementazione di script avanzati e transazioni multi-firma, offrendo maggiore flessibilità. Sono compatibili sia con i vecchi che con i nuovi tipi di indirizzi Bitcoin.

4.2 Portafogli Bitcoin (Wallets)

I portafogli Bitcoin sono strumenti software o hardware che consentono agli utenti di gestire le proprie chiavi private e, di conseguenza, i propri fondi Bitcoin. La scelta del tipo di portafoglio implica un compromesso tra sicurezza e convenienza.

4.2.1 Tipologie di portafogli

- **Hot Wallets:** Questi sono portafogli software che rimangono connessi a Internet, come applicazioni desktop, mobile (es. Coinbase, Trust Wallet, Electrum) o portafogli forniti direttamente dagli exchange. Sono estremamente convenienti per l'uso quotidiano, le transazioni rapide e il trading. Tuttavia, la loro costante connessione alla rete li espone a un rischio maggiore di attacchi informatici, come hacking o malware.
- **Cold Wallets:** Rappresentano la soluzione più sicura per la conservazione di Bitcoin, poiché mantengono le chiavi private completamente offline.
  - **Hardware Wallets:** Dispositivi fisici dedicati (es. Ledger, Trezor) progettati per generare e memorizzare le chiavi private in un ambiente isolato e sicuro. Le transazioni vengono firmate all'interno del dispositivo, senza che la chiave privata lasci mai l'hardware.
  - **Paper Wallets:** Consistono in una copia fisica (stampata o scritta a mano) della coppia di chiavi pubblica/privata. Per garantire la massima sicurezza, queste chiavi dovrebbero essere generate su un dispositivo completamente disconnesso da Internet.

4.2.2 Sicurezza e best practice

La sicurezza ultima dei fondi Bitcoin ricade in gran parte sulla responsabilità dell'utente nella gestione della propria chiave privata e della seed phrase. L'utente non può delegare completamente la sicurezza a terzi, come avviene nel sistema bancario tradizionale. La conoscenza delle migliori pratiche di gestione delle chiavi e la consapevolezza dei rischi associati alle diverse tipologie di portafoglio sono essenziali per prevenire la perdita di fondi. Questo aspetto rafforza il principio di decentralizzazione di Bitcoin, ma impone anche un onere significativo sull'utente.
Le pratiche di sicurezza fondamentali per la gestione dei portafogli Bitcoin includono:

- **Seed Phrase (Frase di Recupero):** Al momento della creazione di un portafoglio, viene generata una "seed phrase" (o frase mnemonica), una sequenza di 12 o 24 parole. Questa frase è il backup fondamentale per recuperare il portafoglio e tutte le chiavi in esso contenute. Deve essere conservata offline, in un luogo estremamente sicuro e protetto (ad esempio, una cassaforte ignifuga), e mai condivisa con nessuno.

- **Non Condividere Mai la Chiave Privata:** La chiave privata è l'equivalente della "password" per i propri fondi. Rivelarla a chiunque significa cedere il controllo completo sui propri Bitcoin.
- **Autenticazione a Due Fattori (2FA):** Abilitare l'autenticazione a due fattori (2FA) su tutti i portafogli e gli exchange che la supportano aggiunge un ulteriore livello di sicurezza, richiedendo una seconda forma di verifica oltre alla password.
- **Verifica Accurata dell'Indirizzo:** Prima di inviare o ricevere Bitcoin, è cruciale verificare attentamente l'indirizzo del portafoglio del destinatario. Errori di battitura o attacchi di "address poisoning" (dove un aggressore invia transazioni a un indirizzo simile per indurre l'utente a copiare quello sbagliato) possono portare alla perdita irreversibile dei fondi.
- **Uso di Indirizzi Diversi:** Per migliorare la privacy, è consigliabile utilizzare un nuovo indirizzo Bitcoin per ogni transazione in entrata. Molti portafogli generano automaticamente nuovi indirizzi dopo ogni transazione.

4.3 HD Wallets (Hierarchical Deterministic Wallets)

Gli HD Wallets (Hierarchical Deterministic Wallets), definiti dal BIP 32, rappresentano un'innovazione significativa nella gestione delle chiavi Bitcoin, migliorando notevolmente l'usabilità e la sicurezza rispetto ai portafogli non-deterministici precedenti.[28]

4.3.1 Principi di generazione gerarchica delle chiavi

- Un HD Wallet genera tutte le sue chiavi e indirizzi da un'unica *"seed phrase" (frase mnemonica).*
- Sono definiti *"gerarchici"* perché le chiavi e gli indirizzi derivati possono essere organizzati in una struttura ad albero, con una chiave master che genera chiavi figlio, che a loro volta possono generare chiavi nipote, e così via.
- Sono *"deterministici"* perché, data la stessa seed phrase, le chiavi e gli indirizzi saranno sempre generati nello stesso modo e nello stesso ordine.

Questo approccio ha rivoluzionato i portafogli di criptovalute, offrendo supporto multi-valuta, semplificando il recupero dell'account e migliorando la sicurezza e la privacy.[29]

4.3.2 Seed Phrase e Derivation Paths

- La **Seed Phrase** è una sequenza di 12 o 24 parole casuali che costituisce la fonte primaria (il "seed" di 64 byte) da cui derivano tutte le chiavi e gli indirizzi del portafoglio. È l'unico elemento che deve essere salvato per il backup dell'intero portafoglio.
- Le chiavi all'interno di un HD Wallet sono generate seguendo specifici **percorsi di derivazione** (derivation paths). Questi percorsi definiscono la struttura gerarchica e il tipo di indirizzi generati. Esempi comuni includono:

- **m/44'/0'/0'** per indirizzi P2PKH (che iniziano con "1").
- **m/49'/0'/0'** per indirizzi P2SH-P2WPKH (che iniziano con "3").
- **m/84'/0'/0'** per indirizzi P2WPKH (che iniziano con "bc1").

Percorso	Tipo di indirizzo	Prefisso tipico
m/44'/0'/0'	Legacy (P2PKH)	1...
m/49'/0'/0'	Nested SegWit (P2SH-P2WPKH)	3...
m/84'/0'/0'	Native SegWit (Bech32)	bc1...

Tabella 4: Esempi di derivation paths in HD Wallets

4.3.3 Vantaggi

Gli HD Wallets offrono numerosi vantaggi che li hanno resi lo standard de facto per la maggior parte dei portafogli moderni:

- **User-friendly:** Sono molto più facili da gestire rispetto ai vecchi portafogli che richiedevano la gestione e il backup individuale di ogni singola chiave privata. Con una sola seed phrase, è possibile recuperare un numero illimitato di chiavi e indirizzi.
- **Miglioramento della Privacy:** Consentono la generazione automatica di un nuovo indirizzo per ogni transazione, rendendo più difficile per gli osservatori esterni collegare tutte le transazioni alla stessa identità dell'utente.
- **Supporto Multi-valuta:** Molti HD Wallets possono generare chiavi e indirizzi per diverse criptovalute dalla stessa seed phrase, semplificando la gestione di un portafoglio diversificato.

Gli HD Wallets sono un esempio significativo di come l'innovazione tecnica non si limiti agli algoritmi crittografici di base, ma si estenda anche all'interfaccia utente e alla gestione delle risorse digitali. La seed phrase, pur essendo un singolo punto di fallimento se compromessa, semplifica enormemente il backup e il recupero di un numero potenzialmente infinito di chiavi. Questo rappresenta un trend che mira a rendere la gestione delle criptovalute più accessibile e meno soggetta a errori umani. L'adozione diffusa di Bitcoin e altre criptovalute dipende non solo dalla loro sicurezza intrinseca, ma anche dalla loro usabilità. Gli HD Wallets hanno giocato un ruolo cruciale nel superare una delle maggiori barriere all'ingresso per gli utenti non tecnici, facilitando una gestione più sicura e pratica delle chiavi e, di conseguenza, dei fondi.

4.4 HD Wallets e Firme a Soglia (Threshold Signatures)

La gestione delle chiavi in Bitcoin si è evoluta per bilanciare la sicurezza teorica del protocollo con le esigenze pratiche degli utenti. Un'evoluzione significativa è stata l'introduzione dei HD Wallets (Hierarchical Deterministic Wallets), che permettono di generare un numero illimitato di chiavi e indirizzi da un'unica "seed phrase". Questo ha reso il backup e il ripristino di un portafoglio infinitamente più semplice per l'utente, ma ha anche introdotto un singolo punto di fallimento: se la seed phrase viene compromessa, tutti i fondi sono a rischio.

Una delle evoluzioni più recenti e importanti riguarda i sistemi di firma avanzata. Le Threshold Signatures (TSS) rappresentano un'alternativa più efficiente e privata al tradizionale portafoglio multi-firma (multi-sig).

Tabella 5: Confronto tra Multi-firma e Threshold Signatures

	Multi-firma (Multi-sig)	**Threshold Signatures (TSS)**
Meccanismo	Richiede che ogni firmatario produca una firma separata.	Utilizza la Multi-Party Computation (MPC) per generare una singola firma.
Trasparenza on-chain	Le firme di ogni partecipante sono visibili on-chain.	La transazione appare come una normale transazione a firma singola.
Vantaggi principali	Elimina il singolo punto di fallimento.	Migliora la privacy, riduce le commissioni, più efficiente.
Svantaggi principali	Costi più elevati e minore privacy a causa dell'ingombro on-chain.	Maggiore complessità di implementazione.
Efficienza on-chain	Bassa (più dati per più firme).	Alta (una sola firma).
Casi d'uso	Conti congiunti, escrow, sicurezza aziendale.	Soluzioni di custodia per istituzioni, wallet avanzati.

A differenza di un portafoglio multi-firma, che lascia una "impronta digitale" visibile sulla blockchain, un TSS, basato sulla Multi-Party Computation (MPC), consente a un gruppo di co-firmatari di collaborare per produrre una singola firma che appare indistinguibile da una firma tradizionale. Ciò riduce le commissioni e migliora notevolmente la privacy, dimostrando come le innovazioni crittografiche non si limitino a migliorare la sicurezza, ma rispondano direttamente a problemi di usabilità, costi e riservatezza.

4.4.1 Custodial vs non-Custodial

Un altro aspetto critico della gestione delle chiavi è la distinzione tra wallet custodial e non-custodial:

- In un wallet **custodial**,una terza parte (spesso un exchange) detiene le chiavi private per conto dell'utente, offrendo convenienza e facilità d'uso, ma al costo di una minore autonomia e di un rischio di controparte.
- Un wallet **non-custodial** dà all'utente il controllo completo e la responsabilità esclusiva sulle proprie chiavi private, ma con la piena responsabilità di gestirle e proteggerle da attacchi o perdita. Questa dicotomia riflette un compromesso fondamentale tra la sicurezza e la praticità, che è un tema centrale nell'evoluzione dell'intero ecosistema delle criptovalute.

Questo concetto si estende all'idea di **Self-Sovereign Identity (SSI)**, un modello in cui un individuo, tramite la crittografia e una blockchain, ha la proprietà e il controllo esclusivo sui propri dati di identità. La SSI permette a un utente di presentare credenziali verificate (es. la prova di avere più di 18 anni) senza rivelare alcuna informazione personale sensibile, utilizzando meccanismi come le Zero-Knowledge Proofs. La SSI rappresenta un cambiamento di paradigma che sposta il potere di gestione dell'identità da autorità centralizzate, come i governi o le aziende, all'individuo stesso.

5 Attacchi alla Blockchain Bitcoin

La sicurezza di Bitcoin è un ecosistema complesso, che va oltre la semplice robustezza degli algoritmi crittografici. Molti attacchi sfruttano vulnerabilità a livello di utente, di rete o di implementazione del protocollo. La resilienza complessiva del sistema deriva dalla capacità di identificare e mitigare le minacce su tutti questi fronti.

5.1 Attacchi di Double Spending

Il double spending è una vulnerabilità critica in cui un utente malintenzionato tenta di spendere la stessa unità di criptovaluta o token blockchain più di una volta. Questo problema è intrinseco a tutte le valute digitali che operano senza un'autorità centrale che impedisca la doppia spesa, e la sua risoluzione è stata una delle innovazioni fondamentali di Bitcoin.

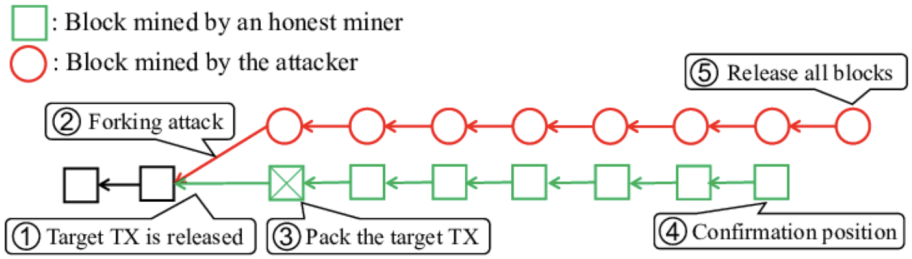

Figura 5: Un diagramma che visualizza come un attaccante possa tentare di spendere la stessa criptovaluta più volte.

5.1.1 Tipologie di Attacchi di Double Spending

- **Race Attack:** Questo attacco si verifica quando un attaccante invia due transazioni quasi simultaneamente con gli stessi fondi. Una transazione è diretta a un commerciante per l'acquisto di beni o servizi, mentre l'altra è inviata a un proprio indirizzo dell'attaccante sulla blockchain. L'obiettivo è che la transazione verso l'attaccante venga confermata per prima dalla rete, invalidando quella al commerciante, che non riceverà il pagamento. Questo tipo di attacco è più probabile che abbia successo se il commerciante accetta transazioni non confermate (infatti viene anche chiamato attacco transazione non confermata).
- **Finney Attack:** Questa è una variante più sofisticata del race attack e richiede la collaborazione di un miner. Un miner malintenzionato crea un blocco che include una transazione in cui i fondi vengono inviati da un *indirizzo A* a un *indirizzo B*, entrambi di proprietà dell'attaccante. Contemporaneamente, l'attaccante invia una seconda transazione (utilizzando gli stessi fondi dell'indirizzo A) a un commerciante (*indirizzo C*). Se il commerciante accetta il pagamento prima che la transazione sia confermata dalla rete, l'attaccante può rilasciare il blocco pre-minato contenente la *transazione A-B*. Questo invalida la *transazione A-C* al commerciante, permettendo all'attaccante di mantenere i fondi.

[30]

5.1.2 Soluzione di Bitcoin al Double Spending

Satoshi Nakamoto(è lo pseudonimo della persona o del gruppo di persone dietro l'invenzione della criptovaluta bitcoin, esistono più teorie riguardanti l'identità di Satoshi Nakamoto ma ad oggi non è ancora chiaro se si tratti di un uomo, una donna o di un gruppo di persone) ha risolto il problema del double spending attraverso un'ingegnosa combinazione di meccanismi crittografici, economici e di consenso :

- **Proof-of-Work (PoW):** l processo di mining richiede una quantità significativa di potenza di calcolo per validare i blocchi e aggiungerli alla blockchain. Questo rende l'atto di creare blocchi (e quindi convalidare transazioni) computazionalmente costoso. Per eseguire un double spend, un attaccante dovrebbe superare la potenza di calcolo della rete onesta per creare una catena alternativa più lunga. Questo costo elevato funge da deterrente economico.
- **Timestamping delle Transazioni:** Le transazioni sono raggruppate in blocchi, ciascuno con un timestamp, e questi blocchi sono concatenati crittograficamente attraverso l'hash del blocco precedente.
- **Consenso Distribuito:** La rete Bitcoin opera secondo la regola che la catena di blocchi più lunga è considerata quella valida. Questo rende estremamente difficile per un attaccante modificare retroattivamente le transazioni in blocchi già consolidati, poiché dovrebbe ricreare l'intera catena successiva con una potenza di calcolo superiore a quella di tutti gli altri miner combinati.
- **Immutabilità:** Una volta che le transazioni sono incluse in un blocco e quel blocco è seguito da un numero sufficiente di blocchi successivi (tipicamente 6 conferme), diventa computazionalmente impraticabile modificarle o annullarle.

La sicurezza di Bitcoin contro il double spending non si basa su un'autorità centrale che "blocca" i fondi, ma su un meccanismo che rende economicamente proibitivo e computazionalmente impraticabile la frode. Questo è un aspetto cruciale del design "trustless" di Bitcoin: la fiducia è riposta nella matematica e negli incentivi economici, non in intermediari.

5.2 Attacco del 51%

L'attacco del 51% è una delle minacce teoriche più temute nelle blockchain basate su Proof-of-Work, come Bitcoin. Si verifica quando un singolo miner o un gruppo di mining riesce ad acquisire il controllo della maggioranza (più del 50%) della potenza di hashing totale della rete.

5.2.1 Meccanismo ed implicazioni

Con la maggioranza della potenza di hashing, l'attaccante otterrebbe un controllo significativo sul ledger e sulla capacità di manipolarlo. Questo gli consentirebbe di:

- **Dettare il Consenso:** L'attaccante potrebbe scegliere quali transazioni includere nei blocchi e quali rifiutare, anche se legittime.

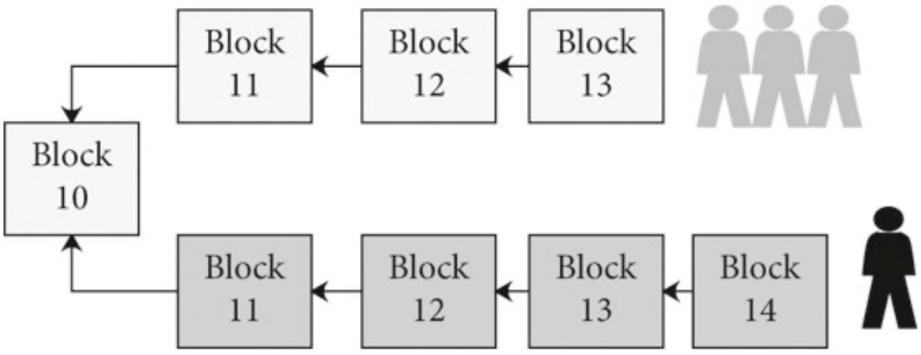

Figura 6: Un diagramma che mostra come un attaccante con la maggioranza della potenza di hashing possa creare una catena privata per manipolare le transazioni.

- **Rifiutare Blocchi Altrui:** Potrebbe impedire ai blocchi minati da altri partecipanti di essere aggiunti alla catena principale, assicurandosi che la competizione non riceva una giusta quota di ricompense.
- **Double Spending:** L'obiettivo principale di un attacco del 51% è il double spending. L'attaccante può creare una propria catena di blocchi "segreta", più lunga di quella pubblica, contenente transazioni che annullano pagamenti precedentemente effettuati e gli restituiscono i fondi. Una volta che questa catena segreta diventa la più lunga e viene trasmessa alla rete, essa sostituisce la catena originale, rendendo nulle le transazioni sulla catena precedente.

[31]

5.2.2 Costi e fattibilità

- **Per Bitcoin:** Un attacco del 51% sulla blockchain di Bitcoin è considerato estremamente improbabile e proibitivamente costoso. Le stime per il costo di un attacco di un solo giorno superano i 15 miliardi di USD. La vasta dimensione e l'elevato hash rate della rete Bitcoin rendono computazionalmente e finanziariamente molto difficile per una singola entità o gruppo superare la potenza di calcolo combinata di tutti gli altri miner onesti.
- **Per Altcoin:** Molte altcoin(altre 'monete'), in particolare quelle con hash rate inferiori e una minore sicurezza di rete, sono significativamente più a rischio di attacchi del 51%.

[32]

5.2.3 Aggiustamento della difficoltà

Ogni 2016 blocchi (circa due settimane), la rete Bitcoin ricalcola il target di difficoltà:

$$D_{nuovo} = D_{vecchio} \times \frac{Tempo_{effettivo}}{2016 \times 600 \text{ sec}}$$

Questo assicura un tempo medio di 10 minuti per blocco, indipendentemente dall'hash power della rete.[52]

5.2.4 Proof-of-Work, autoregolazione e tempo di blocco

Il meccanismo di *Proof-of-Work* (PoW) fa sì che la rete converga su un tempo medio di blocco di circa 10 minuti tramite un **controllo retroazionato** sulla difficoltà. Ogni $N = 2016$ blocchi (circa due settimane), il protocollo calcola il tempo effettivo impiegato T_{eff} per produrli e aggiorna il target di difficoltà D secondo: [55]

$$D_{\text{nuovo}} \;=\; D_{\text{vecchio}} \times \frac{T_{\text{eff}}}{N \cdot 600\ \text{s}}$$

dove 600 s sono 10 minuti. Se i blocchi sono stati trovati troppo *rapidamente* ($T_{\text{eff}} < N \cdot 600$), la difficoltà *aumenta* (il target si abbassa); se troppo *lentamente*, la difficoltà *diminuisce*. In media, con hash power stabile, il tempo atteso resta vicino a 10 minuti.

Dal punto di vista probabilistico, ciascun tentativo di hash è un Bernoulli con probabilità di successo $p = \frac{\text{target}}{2^{256}}$; il numero di tentativi fino al successo è geometrico e, dato un hashrate complessivo H (tentativi/s), il tempo atteso al successo è

$$\mathbb{E}[T] \;=\; \frac{1}{H\,p}.$$

Poiché il retarget *mantiene* $H\,p$ allineato a $1/600$, si ottiene $\mathbb{E}[T] \approx 600$ s.[53]

Autoregolazione economica. La difficoltà non è solo un parametro tecnico: determina il costo energetico per unità di sicurezza. Se il prezzo di BTC cresce, nuovi miner entrano (o i miner esistenti accendono più macchine) perché l'aspettativa di profitto aumenta; l'hashrate H sale, i blocchi arrivano più in fretta e, al retarget successivo, **la difficoltà aumenta**. Questo riallinea i profitti marginali: il costo (energia, ammortamento hardware) per produrre un blocco cresce finché il margine economico si riporta verso l'equilibrio. Viceversa, se il prezzo scende o l'energia rincara, i miner marginali escono, H cala e la difficoltà si riduce. In equilibrio, il sistema:

- mantiene $\mathbb{E}[T] \approx 10$ minuti (controllo tecnico);
- rende *impratiche* le riscritture della catena senza investimenti economici enormi (barriera economica).

Questa è la connessione tra "*impossibilità economica*" degli attacchi (51%, double-spend profondi) e **autoregolazione del PoW**: la difficoltà si adegua endogenamente all'hashrate, trasformando denaro ed energia in sicurezza della rete.[54]

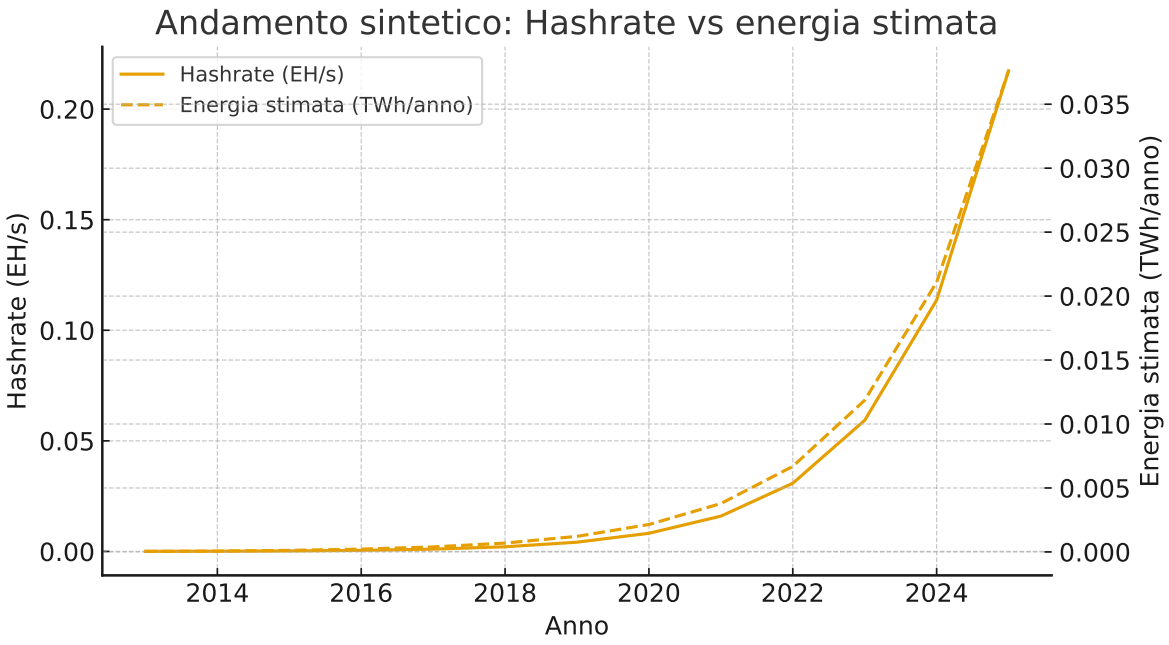

Figura 7: Andamento sintetico (illustrativo) dell'hashrate globale (asse sinistro) e dell'energia stimata (asse destro) nel tempo. L'incremento dell'hashrate è frenato nel lungo periodo dai miglioramenti di efficienza energetica, ma il retarget della difficoltà mantiene il tempo medio di blocco vicino a 10 minuti.

5.2.5 Prevenzioni ed incentivi

La chiave per prevenire gli attacchi del 51% è garantire che nessun singolo miner o gruppo controlli più del 50% della potenza di calcolo della rete. Gli incentivi economici intrinseci al protocollo Bitcoin svolgono un ruolo cruciale: i premi di blocco e le commissioni di transazione spingono i miner a rimanere onesti, poiché investire la propria potenza di calcolo nel mining legittimo è economicamente più redditizio che tentare di frodare la rete. La comunità e il mantenimento della decentralizzazione della potenza di hashing sono quindi fattori cruciali per la sicurezza a lungo termine di Bitcoin.
La sicurezza di Bitcoin, in questo contesto, è il risultato di un equilibrio economico e computazionale. L'attacco del 51% dimostra che anche un sistema decentralizzato basato su Proof-of-Work ha una vulnerabilità teorica legata alla concentrazione di potere computazionale. Tuttavia, la sua inattuabilità pratica per Bitcoin non deriva da un'impossibilità tecnica, ma da un equilibrio economico: il costo dell'attacco supera di gran lunga i potenziali benefici. Questo è un esempio lampante di come la sicurezza di Bitcoin sia intrinsecamente legata alla sua economia e alla distribuzione della potenza di mining. La decentralizzazione della potenza di hashing è quindi un obiettivo continuo e una metrica chiave per la salute della rete, poiché una sua eccessiva centralizzazione potrebbe alterare questo equilibrio economico.

5.3 Altre tipologie di attacchi rilevanti

Oltre agli attacchi di double spending e al 51% attack, la blockchain di Bitcoin e i suoi utenti sono esposti a diverse altre tipologie di minacce, che evidenziano come la sicurezza sia un ecosistema complesso, che va oltre la robustezza degli algoritmi crittografici di base.

- **Attacchi di Phishing:** Questi sono tentativi di frode volti a ottenere le credenziali di un utente, come le chiavi private del portafoglio o le credenziali di accesso agli exchange, tramite l'inganno. Gli attaccanti inviano e-mail o messaggi falsi che sembrano provenire da fonti affidabili, inducendo le vittime a cliccare su link malevoli o a inserire le proprie informazioni su siti web fraudolenti.
- **Attacchi Routing:** In un attacco di routing, gli hacker intercettano i dati mentre vengono trasmessi tra i partecipanti alla blockchain e i provider di servizi Internet. Questo permette loro di estrarre dati riservati o valute senza che i partecipanti alla blockchain si accorgano della minaccia, poiché tutto sembra procedere normalmente.
- **Attacchi Sybil:** Un attacco Sybil si verifica quando un aggressore crea e utilizza un gran numero di false identità di rete (nodi) per sopraffare la rete e interrompere il suo funzionamento o manipolare il consenso. Sebbene le blockchain come Bitcoin siano progettate per resistere a tali attacchi tramite il Proof-of-Work, reti più piccole o con meccanismi di consenso diversi possono essere più vulnerabili.
- **Address Poisoning:** Questo è un attacco emergente in cui un aggressore invia piccole transazioni a un utente, utilizzando un indirizzo generato in modo da essere estremamente simile (ma non identico) a un indirizzo con cui l'utente ha interagito in passato. L'obiettivo è indurre l'utente a copiare e incollare l'indirizzo sbagliato per future transazioni, inviando così inavvertitamente i fondi all'attaccante. La prevenzione richiede una verifica estremamente attenta degli indirizzi di destinazione.
- **Malleabilità delle Transazioni:** Questa è stata una vulnerabilità storica del protocollo Bitcoin, che permetteva a un attaccante di modificare leggermente l'ID di una transazione (TXID) senza alterarne il contenuto effettivo. Se un sistema si basava sul TXID originale per tracciare una transazione, l'attaccante poteva ingannarlo, facendogli credere che la transazione non fosse mai avvenuta o che fosse fallita, potendo potenzialmente ritirare i fondi più volte (come accaduto con l'exchange Mt. Gox).
  - **Soluzione:** La malleabilità delle transazioni è stata risolta con l'implementazione di Segregated Witness (SegWit). SegWit ha "segregato" (separato) i dati della firma (witness data) dai dati principali della transazione, rendendo l'ID della transazione immutabile e non più suscettibile a tali modifiche.

La lista di attacchi dimostra che la sicurezza di Bitcoin non dipende solo dalla robustezza degli algoritmi crittografici di base. Molti attacchi sfruttano vulnerabilità a livello di utente (phishing, address poisoning), di rete (routing, Sybil) o di implementazione del protocollo (malleabilità). La risoluzione della malleabilità tramite SegWit è un esempio di come il protocollo debba evolvere per chiudere queste "falle" non direttamente crittografiche. La sicurezza di un sistema come Bitcoin è un ecosistema complesso che richiede attenzione a molteplici livelli.[35]

6 Privacy in Bitcoin

La privacy in Bitcoin è un argomento complesso e spesso frainteso. Sebbene Bitcoin offra un certo grado di riservatezza, non garantisce l'anonimato completo.

6.1 Pseudonimia vs. Anonimato

Bitcoin è un sistema pseudonimo, non anonimo. Le transazioni avvengono tra indirizzi unici, che sono stringhe alfanumeriche (es *1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa* o *bc1qar0srrr7xfkvy5l643lydnw9re59gtzzwf35f9*) e non sono direttamente collegati a nomi reali o identità personali. Questo aspetto può indurre gli utenti a credere erroneamente che le loro attività siano completamente private.
Tuttavia, esistono limiti significativi alla privacy intrinseca di Bitcoin:

- **Blockchain Pubblica:** Tutte le transazioni sono registrate su una blockchain pubblica e trasparente, accessibile a chiunque nel mondo. Ciò significa che ogni transazione, il suo importo e gli indirizzi di invio e ricezione sono visibili e possono essere tracciati. Questo permette di seguire il percorso dei fondi attraverso la catena di blocchi.
- **Analisi On-Chain:** Anche se gli indirizzi non rivelano direttamente nomi, l'analisi dei dati sulla blockchain, combinata con informazioni esterne, può compromettere la privacy. Ad esempio, gli indirizzi possono essere collegati a individui tramite dati forniti agli exchange (processi KYC - Know Your Customer), indirizzi IP, o attraverso l'analisi di transazioni multi-input che rivelano che più indirizzi appartengono allo stesso proprietario. Strumenti di analisi blockchain possono mappare e raggruppare indirizzi, identificando modelli di spesa e, potenzialmente, l'identità sottostante.
- **Tracciabilità della Storia:** Poiché l'intera cronologia delle transazioni è memorizzata per sempre sulla blockchain, le spese e i saldi di un indirizzo possono essere tracciati nel tempo, rivelando informazioni sulle abitudini finanziarie dell'utente.

Il design di Bitcoin privilegia la trasparenza e la verificabilità (tutte le transazioni sono pubbliche) come meccanismo per garantire il consenso e prevenire la frode (es. double spending). Questa trasparenza è in tensione diretta con il concetto di anonimato. La "pseudonimia" è un compromesso: l'identità è mascherata da un alias, ma l'attività è completamente visibile e potenzialmente collegabile all'identità reale tramite analisi on-chain o dati esterni. Per gli utenti che cercano una maggiore privacy, la comprensione di questa limitazione intrinseca è fondamentale. Non possono fare affidamento solo sul protocollo di base, ma devono attivamente adottare tecniche per migliorare la privacy che aggiungano strati di offuscamento. Questo evidenzia che la privacy in Bitcoin è un obiettivo che richiede uno sforzo proattivo da parte dell'utente.

Pseudonimia vs Anonimato: confronto con sistemi orientati alla privacy In Bitcoin l'identità on-chain è rappresentata da indirizzi (pseudonimi). La piena trasparenza del ledger facilita la verificabilità pubblica, ma espone a tecniche di *clusterizzazione* e *linkability*. Criptovalute orientate alla privacy perseguono un modello differente.

Caratteristica	Bitcoin	Monero / Zcash
Modello di privacy	Pseudonimia (indirizzi pubblici; UTXO trasparenti)	Anonimato forte (ring signatures/stealth addresses in Monero; zk-SNARKs in Zcash)
Trasparenza	Totale: importi e flussi sono pubblici	Limitata: importi e collegamenti offuscati (opt-in in Zcash)
Verificabilità	Semplice, trust-minimized	Più complessa (proof ZK, parametri)
Efficienza	Alta (dati e calcolo ridotti)	Minore (proof più grandi/costose)

Tabella 6: Modelli di privacy: trasparenza di Bitcoin vs approcci orientati all'anonimato [41][42][43] [44]

6.2 Tecniche per migliorare la privacy

Per gli utenti che desiderano migliorare la propria privacy nell'uso di Bitcoin, sono state sviluppate diverse tecniche e protocolli che aggiungono strati di offuscamento alle transazioni on-chain o spostano le transazioni su "secondi strati". La nascita e l'adozione di queste tecniche sono una risposta diretta alle limitazioni di privacy del design originale di Bitcoin (pseudonimia ma trasparenza della blockchain). Queste soluzioni rappresentano un trend di sviluppo di "strati" aggiuntivi che migliorano funzionalità specifiche senza alterare il protocollo di base.

6.2.1 Coin Join

- **Funzionamento:** CoinJoin è una tecnica che aumenta l'anonimato raggruppando i pagamenti di più utenti in una singola transazione. In una transazione CoinJoin, gli input di diversi utenti vengono combinati e mescolati, e gli output vengono distribuiti in modo che sia difficile per un osservatore esterno determinare quale input corrisponda a quale output. Questo offusca la cronologia delle transazioni e riduce la collegabilità tra i fondi.[33]
- **Benefici:** Aumenta l'anonimato, offusca la cronologia delle transazioni, riduce la collegabilità tra indirizzi e migliora la privacy finanziaria, rendendo più difficile per gli strumenti di analisi di rete tracciare le attività individuali. Sebbene non garantisca l'anonimato al 100%, migliora significativamente la privacy rispetto alle transazioni standard.[34]
- **Distinzione da Mixer:** CoinJoin e PayJoin (vedi sotto) mescolano le transazioni direttamente sulla blockchain in modo collaborativo tra gli utenti. I "mixer" o "tumblers", invece, spesso si basano su servizi centralizzati esterni che ricevono i fondi degli utenti, li mescolano con altri e li restituiscono, richiedendo un certo grado di fiducia nel servizio.

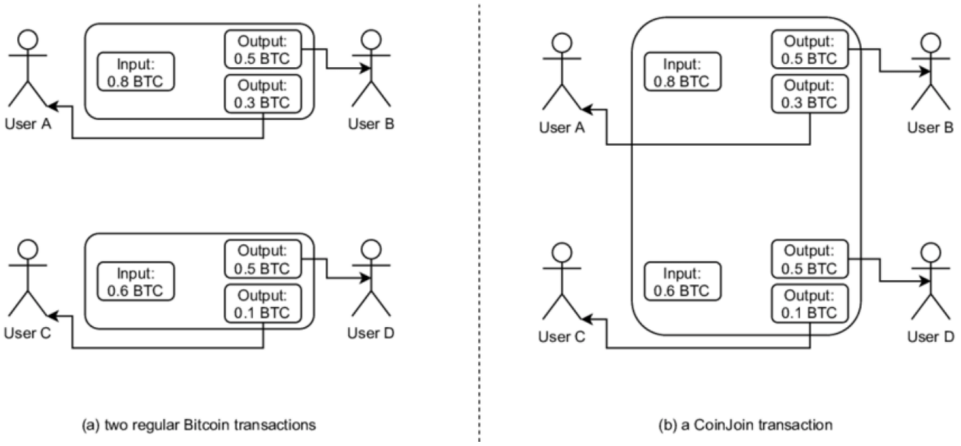

Figura 8: Un diagramma che illustra come più utenti combinino i loro input e output in una singola transazione per aumentare la privacy.

6.2.2 Pay Join

Simile a CoinJoin, PayJoin è una tecnica collaborativa che coinvolge solo due utenti. Entrambi gli utenti combinano i loro fondi nella stessa transazione, facendola apparire come una normale transazione on-chain. Questo rende più difficile per l'analisi della blockchain distinguere tra un semplice pagamento e un'operazione di mescolamento, migliorando la privacy.

6.2.3 Uso di VPN e Tor

- **VPN (Virtual Private Network):** L'utilizzo di una VPN nasconde l'indirizzo IP reale dell'utente e crittografa il traffico Internet. Questo impedisce ai provider di servizi Internet (ISP) e ai siti web di tracciare l'attività online dell'utente e di collegare l'indirizzo IP al suo dispositivo o alla sua posizione fisica.
- **Tor (The Onion Router):** Tor instrada il traffico Internet attraverso una rete di nodi volontari gestiti da utenti in tutto il mondo, nascondendo l'identità dell'utente e rendendo estremamente difficile tracciare l'origine del traffico. Il client Bitcoin Core offre una funzione Tor integrata per migliorare la privacy delle connessioni alla rete Bitcoin.

6.2.4 Lightning Network come soluzione per la Privacy Off-Chain

- **Funzionamento:** Il Lightning Network è un "secondo strato" (Layer-2) costruito sopra la blockchain di Bitcoin. Permette di effettuare un numero elevato di transazioni "off-chain" (fuori dalla blockchain principale) attraverso "canali di pagamento" bidirezionali. Solo l'apertura e la chiusura di questi canali vengono registrate sulla blockchain principale; tutte le transazioni intermedie all'interno del canale rimangono private tra le parti coinvolte.
- **Benefici per la Privacy:** Offre un alto grado di riservatezza per le transazioni frequenti e di piccolo importo, poiché queste non sono pubblicamente visibili sulla blockchain principale, riducendo l'impronta digitale dell'utente.[38][39]

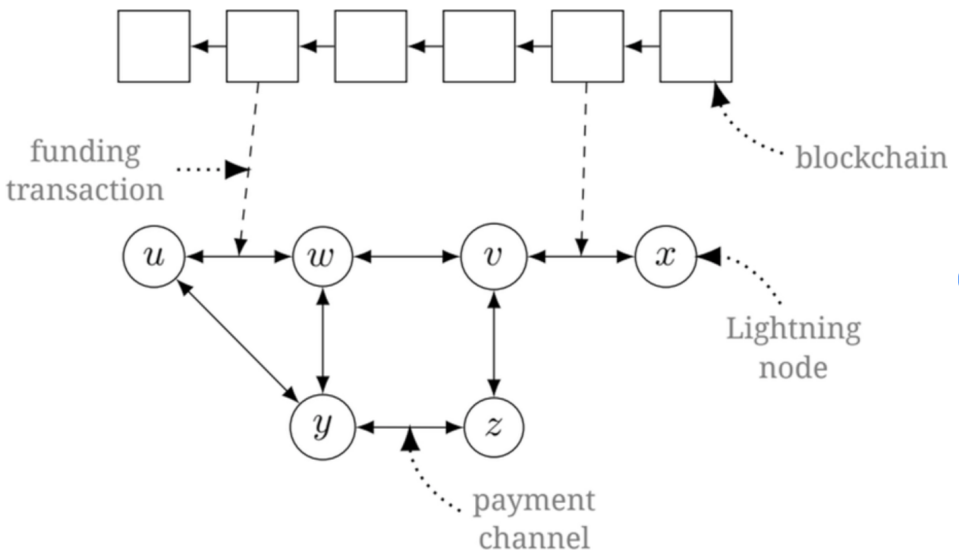

Figura 9: Un diagramma che mostra la struttura dei canali di pagamento e come le transazioni avvengano off-chain.

Il fatto che la privacy non sia *"out-of-the-box"* in Bitcoin ha stimolato l'innovazione. Questo dimostra la flessibilità dell'ecosistema Bitcoin nel costruire soluzioni sopra il layer di base per soddisfare diverse esigenze degli utenti, in questo caso, un maggiore anonimato.

6.2.5 Stonewall e Stonewallx2

Stonewall è una costruzione di spending che mira a rendere ambigua la corrispondenza tra input e output simulando la struttura di una transazione CoinJoin, pur potendo essere eseguita anche da un singolo utente. *Stonewallx2* estende l'idea coordinando due utenti per aumentare il set di anonimato e la non-distinguibilità. Queste tecniche riducono l'efficacia di euristiche come *common-input-ownership*.

Pro e contro

- **Pro:** maggiore ambiguità dei flussi, basso attrito operativo.
- **Contro:** overhead nelle fee e possibili pattern rilevabili da analisti evoluti.

6.2.6 Dandelion++: privacy a livello di rete

La privacy non riguarda solo la struttura della transazione ma anche la sua propagazione sulla rete P2P. *Dandelion++* separa la diffusione in due fasi: *stem* (propagazione lineare casuale) e *fluff* (diffusione a ventaglio). L'obiettivo è rendere più difficile associare l'IP sorgente a una transazione appena creata.[45][46]

6.2.7 Best practice sugli output

La gestione degli output incide direttamente sulla tracciabilità:

- evitare il *re-use* di indirizzi;
- preferire il consolidamento di UTXO in periodi di fee basse;
- bilanciare output di resto (*change*) per non renderli ovvi;
- utilizzare schemi di spending che massimizzino la non-distinguibilità (es. output multipli con importi naturali).

[47]

6.2.8 Metriche di valutazione della privacy

Per confrontare tecniche diverse servono misure quantitative.

Set di anonimato: Se una transazione rende N possibili mittenti (o collegamenti input-output) *indistinguibili*, una misura semplice è:

$$A = \log_2(N) \text{ bit}$$

Esempio: un CoinJoin con 50 partecipanti fornisce circa $\log_2(50) \approx 5.64$ bit.

Entropia di Shannon: Quando le ipotesi non sono equiprobabili, si usa:

$$H = -\sum_{i=1}^{N} p_i \log_2 p_i$$

dove p_i è la probabilità (stimata) che l'ipotesi i sia corretta (es. un certo link input-output). H cresce con l'incertezza del tracciamento.

Linkability: Definiamo la *linkability* come la probabilità $\mathcal{L}$ che due UTXO siano attribuibili allo stesso soggetto dati gli indizi osservati:

$$\mathcal{L} = \Pr\big[\text{stessa entità} \mid \text{metadati, grafi, euristiche}\big].$$

Una buona tecnica di privacy abbassa $\mathcal{L}$ e aumenta A e H.

Tecnica	**Idea**	**Effetto su A, H**	**Effetto su $\mathcal{L}$**
CoinJoin/PayJoin	Mescola input/ridefinisce i ruoli	$A\uparrow$, $H\uparrow$	$\downarrow$
Stonewall(x2)	Ambiguità strutturale degli output	$A\uparrow$	$\downarrow$
Dandelion++	Maschera la sorgente di rete	$-$ (on-chain)	$\downarrow$ (rete)
Lightning (L2)	Routing off-chain onion	$A\uparrow$ (off-chain)	$\downarrow$
Best practice output	Evita pattern linkabili	$H\uparrow$ moderato	$\downarrow$

Tabella 7: Sintesi qualitativa: impatto delle tecniche su set di anonimato (A), entropia (H) e linkability ($\mathcal{L}$).

6.3 Il potenziale rivoluzionario delle Zero-Knowledge Proofs per la privacy in Bitcoin

Il potenziale rivoluzionario delle Zero-Knowledge Proofs (ZKPs) si estende alla privacy. Le ZKPs possono essere utilizzate per creare transazioni "schermate" che nascondono i dettagli sensibili (mittente, destinatario, importo) pur permettendo ai nodi di verificare la loro validità sulla blockchain. Questo concetto è già stato implementato in criptovalute come Zcash. L'applicazione delle ZKPs non si limita alle transazioni: gli **zk-Rollups** aggregano migliaia di transazioni off-chain in un unico batch, generando una singola prova crittografica che viene pubblicata sulla blockchain principale. Questo riduce drasticamente il carico computazionale e i costi, offrendo una soluzione elegante al compromesso tra trasparenza e riservatezza. La sfida è che lo scripting limitato di Bitcoin L1 rende difficile la verifica on-chain di queste complesse prove crittografiche.

SNARK vs STARK: Le *zk-SNARKs* offrono prove corte e verifiche rapide ma, tipicamente, richiedono una *cerimonia di setup fidato*. Le *zk-STARKs* sono *trasparenti* (niente setup fidato) e resistenti ai quanti, ma hanno prove più grandi e verifiche più costose. In Bitcoin, applicazioni pratiche sono più realistiche su *sidechain* o *Layer 2* (rollup a conoscenza zero), mantenendo L1 minimale.[48]

6.4 Self-Sovereign Identity (SSI)

6.4.1 Concetto e Filosofia dell'Identità Auto-Sovrana

La Self-Sovereign Identity (SSI) è un modello di identità digitale che si contrappone radicalmente ai sistemi tradizionali e centralizzati. Invece di affidare la gestione e la custodia dei propri dati personali a grandi entità centralizzate, come governi o giganti della tecnologia (ad esempio, Google o Facebook), l'SSI sposta il controllo totale sull'individuo. L'utente diventa il "sovrano" della propria identità digitale, decidendo in completa autonomia quali dati condividere, con chi e per quale scopo, riducendo drasticamente l'esposizione di informazioni sensibili. Questa filosofia non è solo una scelta tecnologica, ma un profondo cambiamento di paradigma che ha l'obiettivo di eliminare i "single points of failure" e mitigare i rischi di violazione su larga scala dei dati personali, un problema endemico nei modelli di identità centralizzati.

6.4.2 Architettura e Componenti Crittografici dell'SSI

L'architettura dell'SSI si basa su un modello a tre partecipanti, comunemente noto come "Triangolo della Fiducia". I tre ruoli sono interconnessi e fondamentali per il funzionamento del sistema:

- **Issuer (Emittente):** È un'entità, che può essere un'università, un governo o un'azienda, che rilascia una credenziale digitale verificabile (Verifiable Credential) che attesta una certa verità sull'individuo, come un diploma o una patente di guida. L'emittente firma digitalmente questa credenziale per garantirne l'autenticità.
- **Holder (Titolare):** L'individuo che riceve la credenziale dall'Emittente e la memorizza in modo sicuro nel proprio portafoglio digitale. Il Titolare ha il pieno controllo sulla credenziale e può scegliere di presentarla a un verificatore.

- **Verifier (Verificatore):** Un'entità che ha la necessità di verificare la validità di una credenziale presentata dal Titolare. La fiducia del Verificatore nell'Emittente, che può essere stabilita attraverso una blockchain, si trasferisce al Titolare, permettendo la validazione della credenziale.

Per rendere questa architettura tecnicamente realizzabile, l'SSI si avvale di due componenti crittografici principali: i **Decentralized Identifiers (DIDs)** e le **Verifiable Credentials (VCs)**.

- **Decentralized Identifiers (DIDs):** Sono identificatori unici e globali che non dipendono da alcuna autorità centrale. I DIDs sono generati e controllati direttamente dall'utente, che dimostra la proprietà attraverso prove crittografiche come le firme digitali. Spesso, il DID è ancorato a un registro distribuito, come una blockchain, che ne garantisce l'immutabilità e la persistenza. L'uso di DIDs consente all'utente di avere più identità digitali separate per diversi contesti (ad esempio, una per lo shopping online, una per le interazioni professionali) per massimizzare la privacy.[49]
- **Verifiable Credentials (VCs):** Sono l'equivalente digitale e crittografico di documenti fisici come patenti o diplomi. L'Emittente firma digitalmente una VC, che può essere facilmente verificata da un Verificatore. Le VCs sono "manipolazioni evidenti" **(tamper-evident)** per design e possono contenere attributi specifici dell'identità. Un vantaggio fondamentale delle VCs è la possibilità di applicare la "divulgazione selettiva" **(selective disclosure)**, permettendo all'utente di rivelare solo le informazioni strettamente necessarie per una verifica, mantenendo il resto dei dati privato.citew3c-vc-model

Tabella 8: Componenti , Ruoli e Tecnologie del SSI

Componente	**Ruolo nel "Triangolo della Fiducia"**	**Funzione Chiave**	**Tecnologie di Supporto**
Issuer	Emittente della credenziale	Rilascia credenziali firmate digitalmente	Blockchain, crittografia a chiave pubblica
Holder	Custode dell'identità	Memorizza e presenta le credenziali	Digital Wallet, crittografia a chiave pubblica, ZKPs
Verifier	Verificatore della credenziale	Convalida l'autenticità della credenziale	Blockchain, crittografia a chiave pubblica
DID	Ancoraggio dell'identità	Identificatore unico e decentralizzato dell'utente	Blockchain, crittografia a chiave pubblica
VC	Dato dell'identità	Attributi di identità firmati digitalmente dall'emittente	Crittografia a chiave pubblica, Selective Disclosure

L'SSI non è solo un miglioramento tecnico, ma una profonda evoluzione sociale e politica. Sposta il potere di "testimoniare l'esistenza legale" di un individuo dal governo o da un'autorità centrale all'individuo stesso, un cambiamento che ha il potenziale di ridefinire le relazioni tra cittadini, aziende e istituzioni. La sua architettura decentralizzata e l'uso della crittografia offrono un'infrastruttura di fiducia per l'identità che è, per certi versi, analoga all'infrastruttura di fiducia finanziaria creata da Bitcoin.

Standard e interoperabilità: La SSI si appoggia a standard W3C: *Decentralized Identifiers* (DID) e *Verifiable Credentials* (VC). In ambito europeo, eIDAS 2.0 punta a portafogli d'identità digitale; l'integrazione con sistemi non-custodial richiede interfacce che preservino la privacy (*selective disclosure*, proof ZK) e meccanismi di recupero sicuri per l'utente.[51]

6.5 Limiti e sfide future

Resta una tensione strutturale tra **trasparenza** (auditabilità globale) e **privacy** (non-tracciabilità). L1 di Bitcoin privilegia semplicità e verificabilità; ciò suggerisce che innovazioni forti in tema di privacy saranno principalmente:

- **off-chain** (es. Lightning) o in **sidechain** specializzate;
- **opt-in**, con compatibilità retroattiva e impatto nullo sul consenso;
- accompagnate da metriche trasparenti (A, H, linkability) per valutarne l'efficacia.

Il disegno modulare consente di evolvere la privacy senza compromettere la robustezza del livello base.

7 Scalabilità di Bitcoin

La scalabilità di Bitcoin è una delle sfide più significative che la rete deve affrontare per supportare un'adozione di massa. Si riferisce alla capacità del sistema di gestire un volume crescente di transazioni senza compromettere le sue caratteristiche fondamentali.

7.1 Il problema della scalabilità di Bitcoin

Bitcoin, pur essendo un pioniere della tecnologia blockchain, presenta limiti intrinseci alla sua scalabilità sul layer base, che sono il risultato di scelte di design deliberate.

7.1.1 Limiti attuali

- **Dimensione del Blocco:** Il protocollo Bitcoin impone un limite tecnologico di 1 MB per la dimensione di ciascun blocco. Questo limite fu introdotto da Satoshi Nakamoto con l'obiettivo di prevenire l'ingolfamento della rete e mantenere i costi di gestione di un nodo Bitcoin relativamente bassi.
- **Transazioni al Secondo (TPS):** A causa del limite di 1 MB e del tempo medio di creazione di un blocco (circa 10 minuti), la rete Bitcoin può elaborare solo un numero limitato di transazioni, stimato intorno alle 7 transazioni al secondo. Questa capacità è significativamente inferiore rispetto ai sistemi di pagamento tradizionali, come Visa, che può gestire decine di migliaia di transazioni al secondo (circa 65.000 TPS).
- **Congestione della Rete e Commissioni:** Quando la domanda di transazioni supera la capacità della rete, si verifica una congestione. Questo si traduce in tempi di elaborazione delle transazioni più lunghi e in un aumento delle commissioni di transazione, poiché gli utenti sono costretti a "fare un'offerta" più alta per assicurarsi che le loro transazioni vengano incluse rapidamente in un blocco dai miner.

7.1.2 Il "Trilemma della Scalabilità" della Blockchain

Il problema della scalabilità di Bitcoin può essere compreso attraverso il concetto del "trilemma della scalabilità" della blockchain. Questo trilemma suggerisce che una blockchain può ottimizzare solo due dei tre aspetti fondamentali: decentralizzazione, sicurezza e scalabilità. Bitcoin, nella sua architettura di base, ha privilegiato la decentralizzazione e la sicurezza (attraverso il Proof-of-Work e la replicazione massiva dei nodi), sacrificando in parte la scalabilità sul layer base.
Il limite di 1MB della dimensione dei blocchi non è un difetto, ma una scelta di design deliberata di Satoshi Nakamoto. L'obiettivo era prevenire che il database diventasse troppo grande troppo velocemente, mantenendo i costi di gestione di un nodo Bitcoin relativamente bassi e quindi preservando la decentralizzazione della rete. Questo rivela una causalità diretta tra il limite di dimensione del blocco e il mantenimento della decentralizzazione. La "lentezza" di Bitcoin non è un errore, ma il prezzo pagato per garantire che chiunque possa eseguire un nodo completo e verificare la catena, impedendo così la centralizzazione del potere. Questo è un esempio lampante del "trilemma della scalabilità" in azione.[36]

7.2 Soluzioni di scalabilità On-Chain

Le soluzioni di scalabilità "on-chain" mirano a migliorare la capacità di elaborazione delle transazioni direttamente sulla blockchain principale di Bitcoin. Queste soluzioni hanno spesso generato dibattiti significativi all'interno della comunità.

7.2.1 Aumento della dimensione del blocco

- **Concetto:** L'idea più diretta per aumentare il throughput è semplicemente aumentare la dimensione massima dei blocchi, consentendo di includere più transazioni in ciascuno di essi.
- **Esempio (Bitcoin Cash - BCH):** Questa filosofia ha portato a una hard fork di Bitcoin nel 2017, dando origine a Bitcoin Cash (BCH). BCH ha aumentato la dimensione del blocco, inizialmente a 8 MB e successivamente a 32 MB.
- **Implicazioni:** Sebbene un aumento della dimensione del blocco possa incrementare il numero di transazioni per blocco, comporta anche che la blockchain diventi molto più grande nel tempo. Questo può limitare la capacità degli utenti comuni di scaricare e gestire una copia completa della blockchain su hardware standard, potenzialmente portando a una maggiore centralizzazione dei nodi e, di conseguenza, a un rischio per la decentralizzazione della rete. La questione della dimensione del blocco ha causato una significativa controversia e una scissione all'interno della comunità Bitcoin.

7.2.2 Segregated Witness (SegWit)

- **Meccanismo:** Implementato come soft fork nell'agosto 2017 , SegWit (Segregated Witness) è una soluzione più "elegante" per la scalabilità on-chain. Il suo meccanismo consiste nel "segregare" (separare) i dati della firma (chiamati "witness data") dai dati principali della transazione. Questi dati della firma vengono spostati in una sezione estesa del blocco, non inclusa nel limite di 1 MB del blocco base.
- **Impatto sulla Capacità:** Separando le firme, SegWit libera spazio all'interno del blocco di 1 MB per includere un numero maggiore di transazioni effettive. Questo aumenta il throughput complessivo della rete senza aumentare direttamente la dimensione del blocco base.
- **Risoluzione della Malleabilità:** Un beneficio aggiuntivo e significativo di SegWit è la risoluzione del problema della malleabilità delle transazioni. Poiché i dati della firma non fanno più parte del calcolo dell'ID della transazione (TXID), il TXID diventa immutabile, eliminando la possibilità di alterazioni indesiderate.

L'approccio all'aumento della dimensione del blocco (es. Bitcoin Cash) e SegWit rappresentano due filosofie distinte di scalabilità on-chain. L'aumento diretto della dimensione del blocco è più semplice ma ha implicazioni sulla decentralizzazione (costi maggiori per i nodi). SegWit, invece, è una soluzione più "elegante" che ottimizza l'uso dello spazio esistente e risolve un problema di sicurezza (malleabilità), pur essendo più complessa da implementare. Questo ha portato a una scissione della comunità. La scelta di una soluzione di scalabilità on-chain non è puramente tecnica, ma ha profonde implicazioni sociali e politiche per la comunità della criptovaluta. Le decisioni sulla scalabilità riflettono i

valori fondamentali che una rete desidera privilegiare (es. massima decentralizzazione vs. massimo throughput).

7.3 Hard Fork e Soft Fork

7.3.1 Introduzione al Concetto di Fork nel Contesto Blockchain

Le blockchain, come ogni sistema software, richiedono aggiornamenti periodici per evolversi, integrare nuove funzionalità, correggere vulnerabilità o adattarsi a sfide emergenti. Tuttavia, la natura intrinsecamente decentralizzata di reti come Bitcoin complica notevolmente questo processo, che non può essere gestito tramite un'autorità centrale che impone aggiornamenti. In questo ecosistema, ogni modifica al protocollo richiede un consenso da parte dei partecipanti della rete, tra cui miner, sviluppatori e utenti. Un "fork" rappresenta il momento di divergenza del percorso della blockchain a seguito di un cambiamento nelle regole del protocollo. La distinzione fondamentale tra i tipi di fork risiede nella gestione di questa divergenza e nella compatibilità con il software preesistente. Il modo in cui questi aggiornamenti vengono proposti, discussi e attivati è al centro del dibattito sulla governance di Bitcoin, un processo non formalizzato ma che si manifesta attraverso la pressione economica e sociale dei vari attori.

7.3.2 Hard Fork

Un hard fork è un cambiamento radicale nel protocollo di una blockchain che non è retrocompatibile. Questo significa che i nodi che non aggiornano il loro software alle nuove regole non sono più in grado di riconoscere i blocchi validi della nuova catena come tali, rendendoli di fatto "invalidi". L'esito naturale di un hard fork non supportato da tutti i partecipanti è una scissione permanente della blockchain in due reti distinte, ognuna con le proprie regole e la propria storia da quel momento in poi. Le motivazioni dietro un hard fork possono essere diverse, tra cui l'introduzione di nuove funzionalità, la risoluzione di gravi bug di sicurezza, o la necessità di sanare disaccordi profondi all'interno della comunità. L'attivazione di un hard fork richiede un consenso quasi unanime per evitare la scissione e la conseguente nascita di una nuova criptovaluta.

- **Esempio:** Il caso di studio più significativo e emblematico di hard fork nel contesto Bitcoin è la nascita di Bitcoin Cash (BCH) nel 2017. Il dibattito sulla scalabilità della rete e sul limite di 1 MB per blocco, introdotto da Satoshi Nakamoto, aveva generato una profonda spaccatura nella comunità. Una fazione di sviluppatori e miner, sostenendo la necessità di aumentare il throughput per le transazioni, spingeva per un aumento diretto della dimensione del blocco. Poiché non fu possibile raggiungere un accordo consensuale , questo disaccordo sfociò in un hard fork il 1° agosto 2017, dando origine a Bitcoin Cash, che aumentò la dimensione del blocco a 8 MB e poi a 32 MB. Chiunque possedesse Bitcoin al momento del fork si ritrovò a possedere automaticamente anche un'unità di Bitcoin Cash, un evento che ha esemplificato la dinamica di creazione di nuove valute a partire da un hard fork.

- **Implicazioni e rischi:** Le implicazioni e i rischi degli hard fork sono significativi. In primo luogo, la scissione può portare a una profonda frammentazione della comunità e del potere di hashing, con conseguente riduzione della sicurezza della nuova rete. Le nuove catene, dotate di un hash rate inferiore, sono più vulnerabili agli attacchi del 51%. In secondo luogo, un hard fork non gestito correttamente può esporre gli utenti a rischi come gli attacchi di replay, in cui una transazione valida su una catena può essere "riprodotta" sull'altra, causando la perdita di fondi. La storia di Bitcoin Cash e il dibattito sulla scalabilità illustrano in modo chiaro come la scelta di un hard fork sia una manifestazione diretta del "trilemma della scalabilità" (decentralizzazione, sicurezza, scalabilità) in azione. La fazione di Bitcoin Cash ha deliberatamente sacrificato la coesione della comunità, e parte della sicurezza che ne derivava, per perseguire una maggiore scalabilità on-chain, dimostrando che il trilemma non è solo un concetto teorico ma ha conseguenze reali sulla struttura di una rete e sulla sua governance.

7.3.3 Soft Fork

A differenza degli hard fork, un soft fork è un aggiornamento retrocompatibile del protocollo. Le nuove regole sono più restrittive rispetto a quelle precedenti, il che significa che i nodi non aggiornati continueranno a convalidare i blocchi secondo le vecchie regole, ma i nuovi blocchi, che rispettano le regole più stringenti, saranno riconosciuti come validi anche dai vecchi nodi. Questa compatibilità evita una scissione permanente della blockchain, permettendo alla rete di mantenere un'unica catena continua. L'attivazione di un soft fork può essere un processo complesso che richiede una maggioranza di consenso, spesso espressa dai miner, che segnalano la loro disponibilità all'aggiornamento. I meccanismi di attivazione sono evoluti nel tempo, passando da semplici **hardcoded height** a sistemi più sofisticati come **BIP9** e **BIP8**, che cercano di coordinare l'aggiornamento con il minimo rischio di divergenze.
I casi di studio più notevoli di soft fork di successo sono SegWit e Taproot:

- **SegWit (Segregated Witness):** Implementato con successo nel 2017, SegWit è stato una soluzione "elegante" per la scalabilità. Ha risolto la storica vulnerabilità della malleabilità delle transazioni e ha separato i dati delle firme (witness data) dal corpo principale della transazione. Questo ha permesso di includere più transazioni in ogni blocco da 1 MB senza aumentare il limite di dimensione, migliorando il throughput complessivo della rete.
- **Taproot:** Attivato nel novembre 2021, Taproot è stato il più grande soft fork dopo SegWit e ha rappresentato un'importante evoluzione per Bitcoin. Questo aggiornamento ha introdotto le firme Schnorr, più efficienti e meno ingombranti di quelle ECDSA. Ha anche integrato la tecnologia MAST (Merkelized Abstract Syntax Tree), che permette di rendere le transazioni complesse (ad esempio, quelle multi-firma o con smart contract) indistinguibili dalle transazioni semplici on-chain. Il risultato è un significativo miglioramento della privacy, dell'efficienza in termini di spazio e, di conseguenza, della scalabilità del network.

La preferenza storica di Bitcoin per i soft fork è una conseguenza diretta della filosofia del protocollo, che privilegia la continuità della rete e la decentralizzazione rispetto a innovazioni radicali. Il soft fork è intrinsecamente una scelta più sicura che consente di aggiornare il protocollo senza rischiare una scissione, una decisione che riflette l'im-

portanza di preservare la coesione della comunità. Questo approccio ha portato allo sviluppo di soluzioni creative come SegWit e Taproot, che hanno ottimizzato l'uso delle risorse esistenti (come lo spazio nel blocco) invece di modificarle in modo aggressivo, un chiaro segno di come la governance di Bitcoin si manifesti attraverso la negoziazione e l'evoluzione graduale piuttosto che attraverso la rottura.

7.3.4 Analisi Comparativa e di Governance

I fork non sono semplicemente eventi tecnici, ma sono manifestazioni cruciali della governance in una rete decentralizzata. La scelta tra un hard fork e un soft fork riflette un profondo disaccordo ideologico e una valutazione dei rischi associati a ciascun approccio. Un hard fork rappresenta una rottura del consenso e la formazione di un nuovo network, mentre un soft fork è un'evoluzione del consenso stesso, che consente a una rete di adattarsi senza frammentarsi.

Tabella 9: Confronto tra Hard Fork e Soft Fork

Caratteristica	**Hard Fork**	**Soft Fork**
Compatibilità	Non retrocompatibile	Retrocompatibile
Conseguenza	Scissione permanente della blockchain	Singola catena continua
Consenso Richiesto	Quasi unanime per evitare scissioni	Maggioranza (spesso dei miner)
Esempi Noti	Bitcoin Cash (BCH)	SegWit, Taproot
Implicazioni di Governance	Frammentazione della comunità e del potere di hashing	Coesione della comunità e progressi incrementali

7.4 Soluzioni di scalabilità Off-Chain (Layer-2)

Le soluzioni di scalabilità "off-chain", o Layer-2, rappresentano un cambiamento di paradigma nell'approccio alla scalabilità delle blockchain. Invece di modificare il layer di base (che potrebbe compromettere la decentralizzazione o la sicurezza), si costruiscono "strati" aggiuntivi che gestiscono la maggior parte delle transazioni al di fuori della blockchain principale. Questo permette a Bitcoin di mantenere la sua robustezza e decentralizzazione sul layer 1, mentre il layer 2 fornisce la velocità e l'efficienza necessarie per l'adozione di massa.

7.4.1 Lightning Network

- **Concetto:** Il Lightning Network è un protocollo di pagamento di "secondo strato" (Layer-2) che opera sopra la blockchain di Bitcoin. È progettato per abilitare transazioni veloci, economiche e sicure di Bitcoin "off-chain".
- **Canali di Pagamento:** Funziona creando "canali di pagamento" bidirezionali tra due utenti. Per aprire un canale, due utenti depositano una quantità concordata di Bitcoin in una transazione multi-firma sulla blockchain principale. Questa transazione iniziale crea un "ledger" sulla rete Lightning per registrare le transazioni successive tra i due utenti, al di fuori della blockchain di Bitcoin.
- **Transazioni Off-Chain:** Una volta aperto il canale, le parti possono effettuare un numero illimitato di transazioni istantanee e a basso costo tra loro. Ogni transazione viene registrata nel ledger del canale, aggiornando i saldi. Solo l'apertura e la chiusura del canale vengono consolidate in un'unica transazione e registrate sulla blockchain principale di Bitcoin.
- **Routing:** Il Lightning Network non richiede che ogni utente abbia un canale diretto con ogni altro utente. I pagamenti possono essere instradati attraverso una rete di canali interconnessi, in modo simile a come i dati vengono instradati su Internet.
- **Vantaggi:** Offre transazioni quasi istantanee e con costi trascurabili, rendendolo ideale per micropagamenti e transazioni frequenti. Riduce il consumo energetico, poiché non è richiesto mining per ogni transazione all'interno del canale. Fornisce un alto grado di riservatezza, poiché le transazioni all'interno dei canali non sono pubblicamente visibili sulla blockchain principale. Utilizza smart contract e multi-firme per garantire che i fondi raggiungano i destinatari designati.
- **Preoccupazioni:** Alcune preoccupazioni includono la potenziale replicazione di un modello "hub-and-spoke" (centralizzazione dei nodi più grandi), rischi di frode (es. "closed-channel fraud" se un utente chiude un canale in modo disonesto), commissioni di routing e la complessità per gli utenti meno esperti.

[37]

7.4.2 Sidechains

- **Concetto:** Le sidechain sono blockchain separate che possono interagire con la blockchain principale (mainchain), come Bitcoin, in modo bidirezionale. Ciò significa che gli asset possono essere "bloccati" sulla mainchain e poi "sbloccati" sulla sidechain, permettendo il trasferimento di valore tra le due catene.

- **Vantaggi:** Consentono di sperimentare nuove funzionalità, regole o algoritmi di consenso senza modificare il protocollo di base di Bitcoin. Possono offrire maggiore scalabilità o funzionalità specifiche (es. smart contract più complessi) che non sono native sulla mainchain.
- **Implicazioni:** Introducono nuovi rischi di sicurezza, poiché la sicurezza di una sidechain è indipendente da quella della mainchain. Richiedono anche meccanismi di "two-way peg" per garantire il trasferimento sicuro degli asset.

[40]

Tabella 10: Confronto tra Lightning Network e Sidechains

Caratteristiche	**Lightning Network**	**Sidechains**
Obiettivo	Micropagamenti istantanei e a basso costo.	Sperimentare nuove funzionalità e algoritmi di consenso.
Meccanismo di Funzionamento	Crea "canali di pagamento" bidirezionali tra due parti per transazioni off-chain.	Blockchain separate che interagiscono con la mainchain tramite un "two-way peg".
Transazioni	Le transazioni off-chain sono istantanee e non visibili sulla blockchain principale.	Le transazioni avvengono sulla sidechain; solo il trasferimento di asset on-chain è necessario per entrare/uscire.
Livello di Sicurezza	Basato sulla sicurezza della blockchain di Bitcoin; i canali possono essere chiusi sulla mainchain.	La sicurezza della sidechain è indipendente da quella di Bitcoin.
Privacy	Le transazioni all'interno dei canali sono private.	Maggiore riservatezza, ma la sicurezza è legata al meccanismo di peg.
Decentralizzazione	Potenziale rischio di centralizzazione attorno ai nodi di routing "hub".	Può essere meno decentralizzata della mainchain, a seconda dell'implementazione.

Il Lightning Network è un protocollo di Layer 2 che consente di effettuare transazioni quasi istantanee e a costi trascurabili attraverso "canali di pagamento". Solo l'apertura e la chiusura di questi canali vengono registrate sulla blockchain principale. Il routing a cipolla (onion routing) maschera la provenienza e la destinazione dei pagamenti, garantendo un elevato livello di riservatezza.

Le Sidechains sono un'altra soluzione di scalabilità, ovvero blockchain indipendenti che possono interagire con la rete Bitcoin, consentendo l'uso di Bitcoin per funzionalità più avanzate (es. contratti intelligenti complessi) senza modificare il protocollo di base. Questo modello modulare, con il Layer 1 che funge da "strato di liquidazione" sicuro e i Layer 2 che gestiscono il volume, rappresenta un'evoluzione strategica che permette a Bitcoin di mantenere i suoi principi fondanti senza sacrificare la sua capacità di scalare per un'adozione di massa.

L'emergere e l'adozione di Layer-2 come Lightning Network suggeriscono un futuro modulare per le blockchain, dove il layer di base si concentra sulla sicurezza e decentralizzazione, mentre i layer superiori gestiscono la scalabilità e le funzionalità avanzate. Questo approccio permette alla rete di crescere senza sacrificare i principi fondamentali, ma introduce anche nuove complessità e potenziali punti di centralizzazione (es. gli "hub" del Lightning Network) che richiederanno monitoraggio continuo.

8 Conclusioni

L'analisi approfondita della crittografia di base e della sua applicazione in Bitcoin rivela una complessa interazione di principi matematici, innovazioni tecnologiche e dinamiche socio-economiche. La crittografia, con i suoi obiettivi fondamentali di riservatezza, integrità, autenticazione e non ripudio, non è semplicemente uno strumento di sicurezza, ma il pilastro abilitante che ha permesso la nascita e l'evoluzione di un sistema decentralizzato e "trustless" come Bitcoin.
La storia della crittografia, caratterizzata da una costante "corsa agli armamenti" tra cifratura e crittoanalisi, sottolinea che la sicurezza non è mai un punto di arrivo statico. Questa lezione è di vitale importanza per Bitcoin, poiché algoritmi robusti oggi potrebbero affrontare nuove minacce in futuro, come quelle poste dai computer quantistici, rendendo necessaria una continua evoluzione del protocollo.
Gli elementi crittografici specifici di Bitcoin, come l'uso dell'hashing SHA-256 e la struttura del Merkle Tree, non solo garantiscono l'integrità e la verificabilità delle transazioni, ma fungono anche da soluzioni intrinseche per l'efficienza e la scalabilità del sistema. La firma digitale ECDSA è il meccanismo cruciale che traduce la proprietà digitale in un formato crittografico verificabile, ponendo la responsabilità ultima della sicurezza dei fondi nelle mani dell'utente, attraverso la gestione della chiave privata.
La gestione delle chiavi, evoluta dai semplici indirizzi ai complessi HD Wallets, dimostra l'impegno dell'ecosistema Bitcoin nel migliorare l'usabilità senza compromettere la sicurezza. Tuttavia, la consapevolezza e l'educazione degli utenti rimangono essenziali per mitigare i rischi legati a minacce come il phishing o l'address poisoning.
Le sfide della scalabilità e della privacy in Bitcoin evidenziano i compromessi intrinseci nel design di una blockchain pubblica. La scelta deliberata di privilegiare la decentralizzazione e la sicurezza sul layer base ha portato a limitazioni di throughput. Tuttavia, l'innovazione ha risposto con soluzioni on-chain come SegWit e, in particolare, con lo sviluppo di "secondi strati" (Layer-2) come il Lightning Network. Questi strati superiori permettono di gestire un volume elevato di transazioni e di migliorare la privacy, senza alterare i principi fondamentali del layer di base. Questo approccio modulare suggerisce un futuro in cui la blockchain principale mantiene la sua robustezza, mentre i layer superiori offrono la flessibilità e l'efficienza necessarie per un'adozione più ampia.
In sintesi, Bitcoin rappresenta un'applicazione rivoluzionaria della crittografia, che ha ridefinito il concetto di fiducia e di sistema finanziario. La sua resilienza e la sua capacità di adattamento dipendono dalla continua ricerca e implementazione di soluzioni che bilancino sicurezza, decentralizzazione, scalabilità e privacy, in un ecosistema in costante evoluzione.

9 Fonti e Riferimenti

Riferimenti

[1] **Fondamenti di crittografia — Cap. 2 2.1–2.2**. Disponibile: https://aws.amazon.com/it/what-is/cryptography/.

[2] **Fondamenti di crittografia — Cap. 2 2.1–2.2**. https://www.bitpanda.com/academy/it/lezioni/crittografia.

[3] **Storia della crittografia — Cap. 1 1.1**. https://www.ibm.com/it-it/think/topics/cryptography-history.

[4] **Crittografia asimmetrica — Cap. 2 2.2.2**. https://it.wikipedia.org/wiki/Crittografia_asimmetrica.

[5] **Cifratura simmetrica — Cap. 2 2.2.1**. https://www.ibm.com/it-it/think/topics/symmetric-encryption.

[6] **Cifrari classici (Vigenère) — Cap. 1 1.1**. https://it.wikipedia.org/wiki/Cifrario_di_Vigen%C3%A8re.

[7] **SHA-256 (principi) — Cap. 2 2.2.4**. https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm.

[8] **ECDSA, curve ellittiche (introduzione) — Cap. 3 3.4**. https://academy.bit2me.com/it/cos%27%C3%A8-la-curva-ellittica-di-ecdsa/.

[9] **Firme digitali (concetti) — Cap. 3 3.4.1**. https://www.proofpoint.com/it/threat-reference/digital-signature.

[10] **Crittografia asimmetrica (pratica KMS) — Cap. 2 2.2.2**. https://cloud.google.com/kms/docs/asymmetric-encryption?hl=it.

[11] **Crittografia asimmetrica (approfondimento) — Cap. 2 2.2.2**. https://www.ibm.com/it-it/think/topics/asymmetric-encryption.

[12] **Crittografia post-quantum (Italia, ACN) — Cap. 2 2.3**. https://www.acn.gov.it/portale/documents/20119/85999/ACN_Crittografia_Quantum_Safe.pdf.

[13] **Standard PQC (NIST) — Cap. 2 2.3**. https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards.

[14] **Blockchain (definizione) — Cap. 3 3.1**. https://it.wikipedia.org/wiki/Blockchain.

[15] **Bitcoin e blockchain (overview) — Cap. 3 3.1**. https://www.pwc.com/us/en/industries/financial-services/fintech/bitcoin-blockchain-cryptocurrency.html.

[16] **Whitepaper Bitcoin spiegato — Cap. 3 3.1**. https://www.bitpanda.com/academy/it/lezioni/il-whitepaper-di-bitcoin-spiegato-in-maniera-semplice.

[17] **Bitcoin (come funziona) — Cap. 3 3.1**. https://www.ig.com/it/bitcoin/cosa-e-bitcoin-e-come-funziona.

[18] **Blockchain: cosa è/cosa non è — Cap. 3 3.1**. https://www.ictsecuritymagazine.com/articoli/blockchain-cosa-cosa-non/.

[19] **Merkle tree (spiegazione semplice) — Cap. 3 3.3.2**. https://medium.com/coinmonks/merkle-tree-a-simple-explanation-and-implementation-48903442bc08.

[20] **Bitcoin: algoritmo di hashing dei blocchi — Cap. 3 3.3.1**. https://cryptohead.io/what-hashing-algorithm-does-bitcoin-use-to-hash-blocks/.

[21] **Funzioni hash (tecnica) — Cap. 2 2.2.3; Cap. 3 3.3.1**. https://learnmeabitcoin.com/technical/cryptography/hash-function/.

[22] **Merkle proof — Cap. 3 3.3.3**. https://crypto.com/glossary/it/merkle-proof.

[23] **Merkle tree/merkle root (articolo) — Cap. 3 3.3.2**. https://bitcoin-in-action.medium.com/merkle-tree-e-merkle-root-bitcoin-libro-8e61af75b908.

[24] **Merkle root (definizione) — Cap. 3 3.3.2**. https://www.investopedia.com/terms/m/merkle-root-cryptocurrency.asp.

[25] **ECDSA (wiki tecnica) — Cap. 3 3.4.1–3.4.5**. https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm.

[26] **Firme in crypto (primer) — Cap. 3 3.4.1**. https://crypto.com/it/university/a-primer-on-digital-signatures-in-cryptocurrency.

[27] **Sicurezza blockchain (IBM) — Cap. 5 5.1–5.3**. https://www.ibm.com/it-it/topics/blockchain-security.

[28] **HD Wallets — Cap. 4 4.3**. https://learnmeabitcoin.com/technical/keys/hd-wallets/.

[29] **HD Wallets (overview) — Cap. 4 4.3**. https://www.gemini.com/cryptopedia/hd-crypto-wallets-hierachichal-deterministic.

[30] **Double spending (glossario) — Cap. 5 5.1**. https://www.crypto.com/glossary/double-spending.

[31] **Attacco del 51% (spiegazione) — Cap. 5 5.2**. https://www.bitpanda.com/academy/en/lessons/what-is-a-51-attack-and-how-is-it-prevented.

[32] **Attacco del 51% (tecnico) — Cap. 5 5.2**. https://learnmeabitcoin.com/technical/blockchain/51-attack/.

[33] **CoinJoin (introduzione) — Cap. 6 6.2.1**. https://www.nadcab.com/blog/bitcoin-coin-join.

[34] **CoinJoin (Trezor) — Cap. 6 6.2.1**. https://trezor.io/learn/advanced/Blockchain-architecture-technologies/what-is-coinjoin.

[35] **Address poisoning — Cap. 5 5.3**. https://it.cointelegraph.com/news/jameson-lopp-sounds-alarm-bitcoin-address-poisoning.

[36] **Scalabilità: problema — Cap. 7 7.1–7.1.2**. https://it.cointelegraph.com/explained/bitcoin-scaling-problem-explained.

[37] **Layer-2 (panoramica) — Cap. 7 7.4**. https://crypto.com/it/university/what-are-layer-2-scaling-solutions.

[38] **Lightning Network (spiegazione) — Cap. 7 7.4.1; Cap. 6 6.2.4**. https://calebandbrown.com/blog/bitcoin-lightning-network-explained/.

[39] **Lightning Network (definizione) — Cap. 7 7.4.1; Cap. 6 6.2.4**. https://www.investopedia.com/terms/l/lightning-network.asp.

[40] **Scalabilità: sidechain e payment channels — Cap. 7 7.4.1–7.4.2**. https://academy.binance.com/it/articles/blockchain-scalability-sidechains-and-payment-channels.

[41] **Monero: ring signatures — Cap. 6 6.1**. https://www.getmonero.org/resources/moneropedia/ringsignatures.html.

[42] **Monero: stealth addresses — Cap. 6 6.1**. https://www.getmonero.org/resources/moneropedia/stealthaddress.html.

[43] **Zcash (tecnologia) — Cap. 6 6.3**. https://z.cash/technology/.

[44] **ZK-SNARKs (Zcash) — Cap. 6 6.3**. https://z.cash/learn/what-are-zk-snarks/.

[45] **Dandelion++ (paper) — Cap. 6 6.2.6**. https://arxiv.org/abs/1805.11060.

[46] **Dandelion++ (approfondimento) — Cap. 6 6.2.6**. https://bitcoinops.org/en/topics/dandelion/.

[47] **Privacy (best practice) — Cap. 6 6.2.7–6.2.8**. https://en.bitcoin.it/wiki/Privacy.

[48] **STARKs (panoramica) — Cap. 6 6.3**. https://starkware.co/stark/.

[49] **DID Core (W3C) — Cap. 6 6.4.2**. https://www.w3.org/TR/did-core/.

[50] **Verifiable Credentials (W3C) — Cap. 6 6.4.2**. https://www.w3.org/TR/vc-data-model/.

[51] **eIDAS (UE) — Cap. 6 6.4**. https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation.

[52] **Mining: difficoltà (wiki) — Cap. 5 5.2.3–5.2.4**. https://en.bitcoin.it/wiki/Difficulty.

[53] **Mining (dev guide) — Cap. 5 5.2.3–5.2.4**. https://developer.bitcoin.org/devguide/mining.html.

[54] **Economia del mining — Cap. 5 5.2.5**. https://www.coinmetrics.io/bitcoin-mining-economics/.

[55] **Mining difficulty (approfondimento) — Cap. 5 5.2.3–5.2.4**. `https://bitcoinops.org/en/topics/mining-difficulty/`.