
UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO DI INGEGNERIA
ELETTRICA ELETTRONICA E INFORMATICA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

---

*Luigi Fontana*

# SOLUZIONE DI ANALISI PER BLOCKCHAIN BITCOIN

Relatore:
Chiar.mo Prof. Davide Patti

---

Anno Accademico 2022/2023

Luigi Fontana

# Soluzione di analisi per Blockchain Bitcoin

*Master's Degree Thesis*

UNIVERSITY OF CATANIA

April 2024

*«The root problem with conventional currency is all the trust that's required to make it work. The central bank must be trusted not to debase the currency, but the history of fiat currencies is full of breaches of that trust.»*

Satoshi Nakamoto

# SOMMARIO

Le Blockchain sono sistemi di archiviazione dati che utilizzano una struttura a catena per registrare informazioni in modo sicuro, trasparente e decentralizzato. I dati così memorizzati sono immutabili e permanenti, poichè l'unione di diverse soluzioni e tecnologie crea una struttura resiliente ed autonoma. L'analisi on-chain, o blockchain analysis, è il processo di raccolta, analisi e interpretazione dei dati presenti nelle blockchain. Questo processo viene effettuato per monitorare le transazioni, per comprendere le attività della rete e per individuare anomalie o utilizzi impropri della tecnologia. Difatti, si pone come strumento cruciale per il riconoscimento di attività criminali che adottano l'utilizzo delle criptovalute. Esistono diverse società private e governative che svolgono questo compito, tuttavia le tecnologie e gli algoritmi utilizzati sono protetti dal diritto d'autore. Inoltre si avverte la mancanza di software Open Source e gratuiti a tal fine. Pertanto, l'obiettivo della tesi è quello di presentare una soluzione fresca ed innovativa che getti le basi per la creazione di software specializzato nell'analisi on chain che sia aperto, libero e trasparente.

# Indice

*Ai miei cari familiari. . .*

# Introduzione

La presente tesi affronta il tema della blockchain analysis, un campo di studio di grande rilevanza nell'ambito delle blockchain. L'obiettivo è quello di fornire un contributo al tema offrendo una soluzione Open Source e accessibile che possa essere utilizzata da ricercatori, sviluppatori e appassionati. Attraverso un'analisi delle metodologie e delle tecniche utilizzate, si intende fornire un contributo originale alla conoscenza del tema, approfondendo in particolare la tecnica nota come Transaction Graph Analysis. La tesi si articola in cinque capitoli:

- Il primo fornisce una panoramica generale sul tema blockchain, con una particolare attenzione alla genesi, evoluzione e tecniche crittografiche implementate.
- Il secondo capitolo presenta nel dettaglio il protocollo Bitcoin, la prima blockchain, approfondendo la tecnologia nel suo funzionamento.
- Il terzo capitolo l'illustra l'analisi on chain, con una panoramica sulle problematiche relative all'utilizzo scorretto delle blockchain e una prospettiva sulle tecniche utilizzate per mitigare il problema.
- Nel quarto viene presentata una soluzione innovativa al problema, con un software sviluppato per fornire uno strumento per l'analisi on chain.
- Infine, nel quinto capitolo si presenta l'applicazione del software su un caso di furto reale, sintetizzando i risultati ottenuti e aprendo a nuovi possibili sviluppi futuri.

# Capitolo 1

# Genesi ed Evoluzione

## 1.1 Il movimento Cypherpunk

Un cypherpunk è un individuo che sostiene l'uso diffuso delle tecnologie come mezzo per raggiungere una società libera, autonoma e basata sulla collaborazione e responsabilità individuale. I cypherpunk, nati negli anni '80 e '90, erano un movimento di attivisti e tecnici uniti per promuovere la privacy e la sicurezza online [1].
Erano contrari alla censura delle informazioni da parte di governi ed aziende e sostenevano il libero accesso ad idee e conoscenze. Hanno contribuito allo sviluppo della crittografia moderna, alla diffusione del software libero e alla promozione della libertà di parola e di privacy nell'era digitale.
Alcuni cypherpunk degni di nota sono:

- **Marc Andreessen:** Cofondatore di Netscape, gli sviluppatori di SSL.
- **Philip Zimmermann:** Creatore di PGP, acronimo di Pretty Good Privacy.
- **Richard Stallman:** Fondatore della Free Software Foundation.
- **Julian Assange:** Fondatore di WikiLeaks.

I cypherpunk sognavano, un sistema di scambio di denaro libero dal controllo dei governi, decentralizzato, anonimo nei riguardi dei dati personali, trasparente nel suo funzionamento e resistente alla censura. Le criptovalute come Bitcoin, Ethereum e Zcash sono state create seguendo queste idee.

## 1.2 La nascita delle Blockchain

Le radici del paradigma tecnologico affondano in diversi decenni di ricerche e innovazioni in campi come la crittografia, l'informatica e la teoria dei sistemi distribuiti [2].
È il 2008 l'anno in cui avviene la pubblicazione del white paper di Bitcoin da parte di Satoshi Nakamoto che introduce la prima criptovaluta basata su blockchain. All'interno viene descritta una soluzione di rete peer-to-peer che consente pagamenti online con una versione totalmente digitale del contante elettronico, che non necessita la presenza di un'istituzione finanziaria e che risolve il problema della doppia spesa [3].
La rete effettua il timestamp delle transazioni monetarie eseguendo l'hash di una catena crescente di blocchi, tramite la proof of work. Questo, crea un registro distribuito che non può essere cambiato senza ricalcolare la proof-of-work. Il registro distribuito dunque, rende la realtà delle transazioni resiliente a manomissioni, poichè il cambiamento di un blocco porterebbe alla corruzione di tutti i blocchi a lui successivi. I nodi della rete sono a conoscenza della storia passata e seguono la realtà basandosi sulla catena che conta il maggior numero di blocchi.
Quando un blocco storico viene modificato, ad esempio per rubare del denaro, gli autori devono convicere l'intera rete della nuova realtà, creando una catena parallela che conti un numero di blocchi maggiore rispetto a quella originale. Questo è un problema non banale data la continua crescita della blockchain e i costi computazionalmente elevati necessari al calcolo della proof-of-work. Oltre all'immutabilità e alla decentralizzazione, la Blockchain offre una soluzione per un sistema che sia:

- **Pseudoanonimo:** Tutte le transazioni e gli indirizzi Bitcoin sono pubblicamente visibili, ma non connessi a dati personali.
- **Permission-less:** Chiunque abbia un accesso ad Internet può partecipare alla blockchain, senza la necessità di ottenere un'autorizzazione.

- **Trasparente:** Le regole ed i codici Bitcoin sono Open-Source. Disponibili pubblicamente per chiunque voglia visualizzarle, studiarle, modificarle o distribuirle.
- **Trustless:** Non esiste fonte di fiducia o autorità di controllo. Le garanzie sono by-design, grazie ai meccanismi di consenso.

Le principali differenze tra le diverse blockchian include, lo scopo per cui sono state progettate, il protocollo di consenso utilizzato, le caratteristiche e le applicazioni. Mentre Bitcoin è principalmente una riserva di valore digitale e un sistema di pagamento, Ethereum è una piattaforma per applicazioni decentralizzate e smart contract, mentre Zcash è progettato specificamente per concentrarsi sulla privacy degli utenti.

## 1.3 Le Criptovalute

Negli ultimi anni, il termine criptovaluta è divenuto sempre più utilizzato in ambienti finanziari, piani aziendali e titoli di giornale. Viene spesso associato ad un'attività criminale del cosiddetto "dark web", ma in realtà, la parola, il concetto ed i prodotti sono entrati a pieno titolo nella coscienza mainstream. Il termine "cripto" deriva dal greco antico "kryptos", che significa "nascosto", "segreto" o "coperto". Nel contesto delle criptovalute, il termine "cripto" si riferisce a valute che utilizzano la crittografia per proteggere le transazioni e garantire la sicurezza del sistema. La parte crittografica costituisce la base per autenticare il possesso della moneta. Infatti, le informazioni sulle transazioni non sono un segreto, ma al contrario, ogni transazione può essere letta da tutti. Le tecniche crittografiche vengono utilizzare per dimostrare la legittima proprietaria del denaro, o meglio, permettono di dimostrare la legittima proprietà di ciò che una transazione ha prodotto. Bitcoin, utilizza una miscela di Hashing SHA256 e Crittografia a curva ellittica.

### 1.3.1 Hashing

Il funzionamento dell'hashing è molto semplice da spiegare; tuttavia, la matematica sottostante è significativamente più complessa.
È sufficiente prendere una parola, un file o un disco rigido, processarli attraverso l'algoritmo di hashing e come risultato si avrà sempre una stringa di

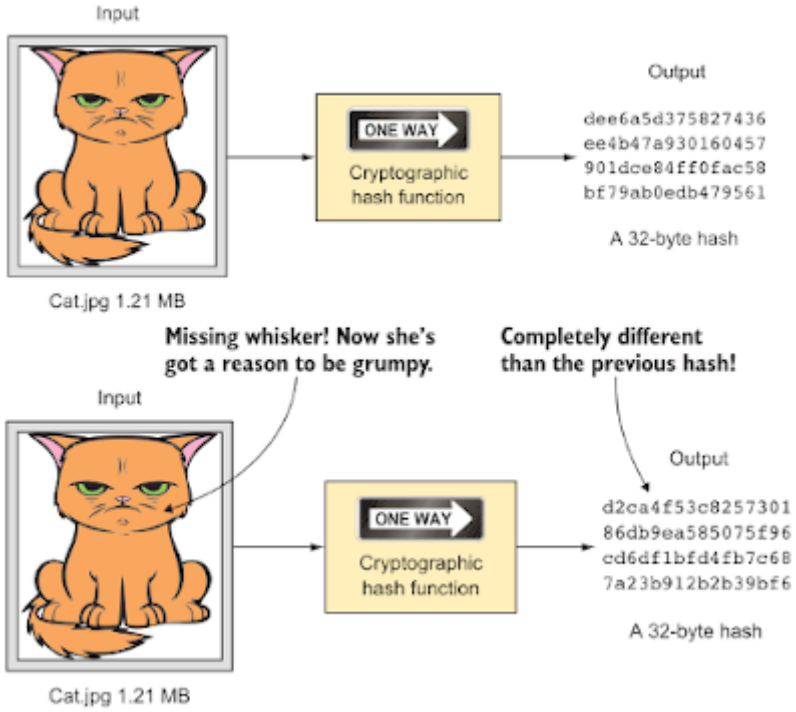


FIGURA 1.1: Esempio algoritmo SHA256

lunghezza fissa. Modificare un solo bit nel file di ingresso porterebbe ad avere un hash risultante completamente diverso, figura 1.1.
Ciò che rende potente questo processo sono le caratteristiche che lo contraddistinguono [4]:

- **Preimage resistence:** proprietà di unidirezionalità, ovvero è possibile generare un hash dato un messaggio, ma praticamente impossibile generare un messaggio dato un hash.
- **Second preimage resistence:** proprietà che garantisce che non sia semplice trovare un messaggio alternativo con lo stesso hash di un determinato messaggio.
- **Collision resistence:** proprietà che si riferisce alla difficoltà di trovare due input distinti che producano lo stesso hash.

Inoltre, SHA256, come tutti gli altri algoritmi di hashing, è computazionalmente poco oneroso, il che consente una popolazione di rete eterogenea composta da una vasta gamma di nodi con potenze computazionali che variano notevolmente.

Le funzioni hash vengono tipicamente utilizzate per la verifica dell'integrità del dato, l'autenticazione, la ricerca e l'archiviazione di informazioni, più in generale, sono fondamentali per la creazione di un'infrastruttura di sicurezza informatica solida e affidabile.
In ottica blockchain sono utilizzate per:

- **La creazione della Blockchain:** Ogni blocco contiene l'hash del blocco precedente. Questo crea una catena che garantisce integrità ed immutabilità. I nodi miners competono per trovare un hash con determinate caratteristiche al fine di portare a termine la proof-of-work.
- **La verifica delle transazioni:** Avviene tramite hashing, che genera un identificativo univoco per ciascuna transazione. I nodi della rete possono garantirne l'integrità calcolando l'hash e confrontandolo con quello presente nel blocco corrispondente.
- **Gli indirizzi Bitcoin:** Sono generati da un hash processato sulla chiave pubblica dell'utente. L'dentificativo univoco risultante consente di ricevere i Bitcoin in modo sicuro ed anonimo, proteggendo la chiave pubblica.
- **Gli script di firma:** controllano come le criptovalute possano essere spese. L'hash è una componente fondamentale della firma digitale, utilizzata per confermare il possessore della moneta.

Dunque sono una componente fondamentale di Bitcoin poichè svolgono un ruolo cruciale in tutti gli aspetti della tecnologia e senza il loro contributo l'intera blockchain non potrebbe esistere.

### 1.3.2 Crittografia a curva ellitica

La crittografia è la scienza e l'arte di trasformare le informazioni in un codice incomprensibile. Questo processo è chiamato cifratura. La decifratura è il processo inverso, trasforma il codice criptato in informazioni leggibili. Ipotizzando che Bob voglia inviare X bitcoin ad Alice, la crittografia a curva ellittica permette di rispondere a due domande fondamentali:

- Come garantire che sia Alice il recettore del denaro?
- Come garantire che sia Bob il committente del denaro?

La crittografia simmetrica, detta anche a chiave segreta, utilizza la stessa chiave per cifrare e decifrare le informazioni, figura 1.2. La chiave va condivisa tramite un canale sicuro e mantenuta segreta, poichè chiunque ne entri in possesso sarà in grado di leggere e modificare i messaggi.

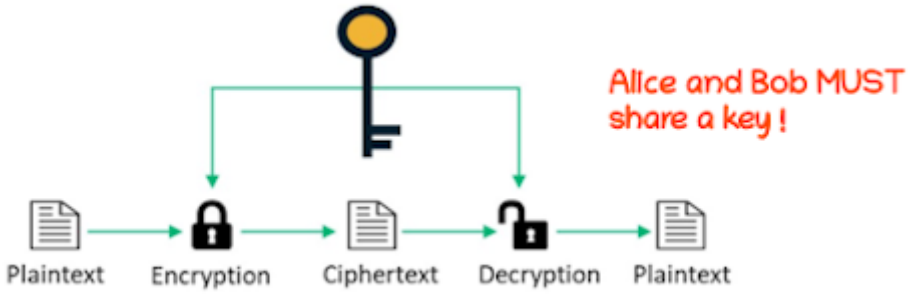


FIGURA 1.2: Esempio di cifratura simmetrica

La crittografia asimmetrica, detta anche a chiave pubblica, si differenzia per l'applicazione di due chiavi: una pubblica e una privata, figura 1.3.
La chiave pubblica viene resa nota a tutti mentre la chiave privata rimane segreta.


FIGURA 1.3: Esemprio di cifratura asimmetrica

Esiste una relazione matematica tra la chiave pubblica e quella privata che consente di utilizzare la chiave privata per generare firme sui messaggi.

Queste firme possono essere convalidate rispetto alla chiave pubblica senza rivelare la chiave privata [5].
La cifratura a curva ellittica o ECC è un paradigma asimmetrico che non viene utilizzato per crittografare, su Bitcoin, bensì per la capacità di generare firme digitali. Una chiave privata può essere applicata a una transazione per produrre una firma digitale. Questa firma può essere generata solo da qualcuno che conosce la chiave privata, tuttavia chiunque abbia accesso alla chiave pubblica e alla transazione può verificarne la firma, figura 1.4
Questa utile proprietà consente a chiunque di ispezionare ogni firma su ogni transazione, garantendo che solo i proprietari delle chiavi private possano produrre le firme valide, necessarie per spendere la criptovaluta [6].
Un recettore di bitcoin, come Alice, percepisce la valuta su una delle sue chiave pubbliche in una transazione firmata dal commitente, Bob. I bitcoin che sta trasferendo Bob sono stati ricevuti su una delle sue chiavi pubbliche, in una transazione precedente, e grazie all'uso della chiave privata corrispondente, può cedere il denaro generando la firma digitale necessaria per sbloccare i bitcoin e trasferirli ad Alice. I nodi della rete possono verificare che la firma di Bob si impegna a generare un output connesso alla chiave pubblica di Alice e quando la transazione viene inclusa all'interno di un blocco, la modifica del proprietario diventa permanente.

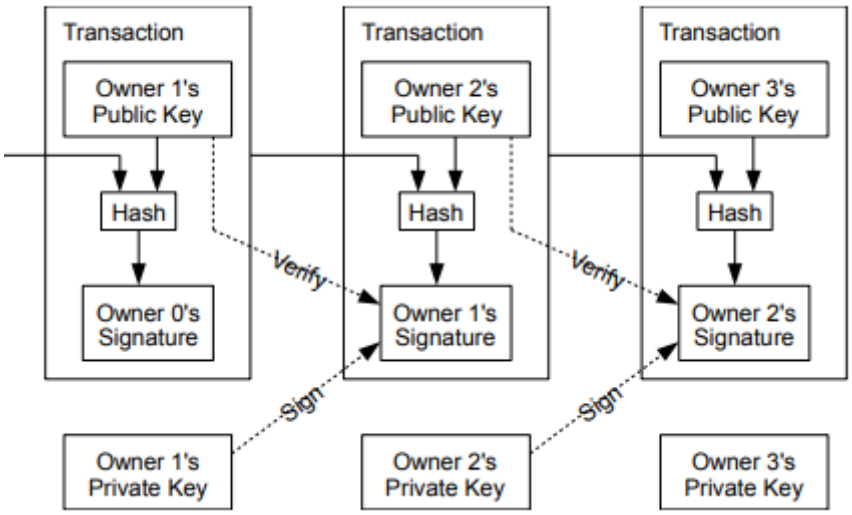


FIGURA 1.4: La catena di transazioni dal paper Bitcoin originale.

## 1.4 Prospettive ed applicazioni

Bitcoin e le Blockchain non producono solamente delle risorse digitali.
Il livello base rappresenta una serie di eventi che non possono essere cambiati, su cui operano diversi attori che puntano sul consenso. Non fidandosi l'uno dell'altro, si affidano a leggi matematiche. Sopra il livello base, è possibile creare delle astrazioni di livello superiore. Un esempio è la Proof-of-Existence (PoE), la quale consente di dimostrare l'esistenza di un documento o di un file digitale in un determinato momento. Utilizzando la blockchain, è possibile registrare un hash crittografico del documento o del file, creando una prova matematica che il documento esisteva in quel momento specifico, senza la necessità di conservare il documento stesso sulla blockchain, figura 1.5.

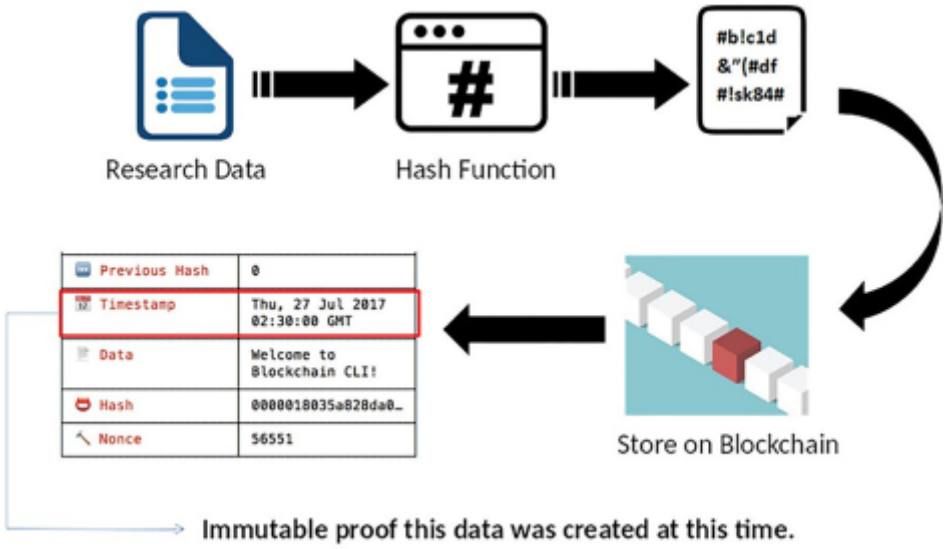


FIGURA 1.5: Esempio di Proof-of-Existence.

Un altro esempio sono gli NFT (Non-Fungible Tokens) che rappresentano un tipo di asset digitale unico, qualcosa di specifico e individuale, come opere d'arte, video o altri beni. Gli NFT possono essere creati e scambiati sopra il livello base della blockchain, sfruttando le sue caratteristiche di immutabilità e sicurezza. In generale, gli utilizzi potenziali sono innumerevoli e si estendono in domini come la finanza, i pagamenti transfrontalieri, la tracciabilità dei prodotti, le catene di approvigionamento (supply chain), la gestione di dati sanitari, l'dentità digitale, il voto digitale, i videogiochi e molto altro.

# Capitolo 2

# Il protocollo Bitcoin

## 2.1 Comprendere la tecnologia

La blockchain è una struttura dati che consiste in un registro di transazioni condiviso da nodi di una rete distribuita. Le transazioni sono all'interno di blocchi connessi tra loro tramite hash crittografico. Ogni blocco presenta delle informazioni che sottolineano le sue caratteristiche. La catena così creata esprime la totalità dei movimenti monetari degli utenti e rappresenta la storia finanziaria della rete. Le transazioni fresche vengono memorizzate in un'area chiamata mempool, comune a tutti i nodi completi, in attesa di essere incluse in un nuovo blocco. La creazione di un nuovo blocco avviene grazie ad un processo chiamato "mining" che aumenta le dimenzioni della catena, modificando permanentemente la realtà finanziaria. Gli utenti, infine, possono utilizzare degli strumenti, chiamati wallet, per memorizzare, gestire e utilizzare le proprie chiavi private e pubbliche con lo scopo di facilitare la gestione dei fondi e l'interazione con la blockchain.

## 2.2 Architettura della rete

Come anticipato la rete Bitcoin è uno specifico tipo di rete distribuita, ovvero una rete Peer-to-Peer o P2P. Una rete distribuita è un sistema in cui risorse e compiti sono ripartiti su più computer anziché essere centralizzati in un solo server. Ciò rende la rete più affidabile, sicura e resiliente a guasti, poichè le risorse, il traffico e le componenti sono divise e ridondate su più nodi. Le caratteristiche che contraddistinguono le reti P2P sono la paritarietà e l'assenza di autorità. I nodi non sono disposti in maniera gerarchica sotto forma di client

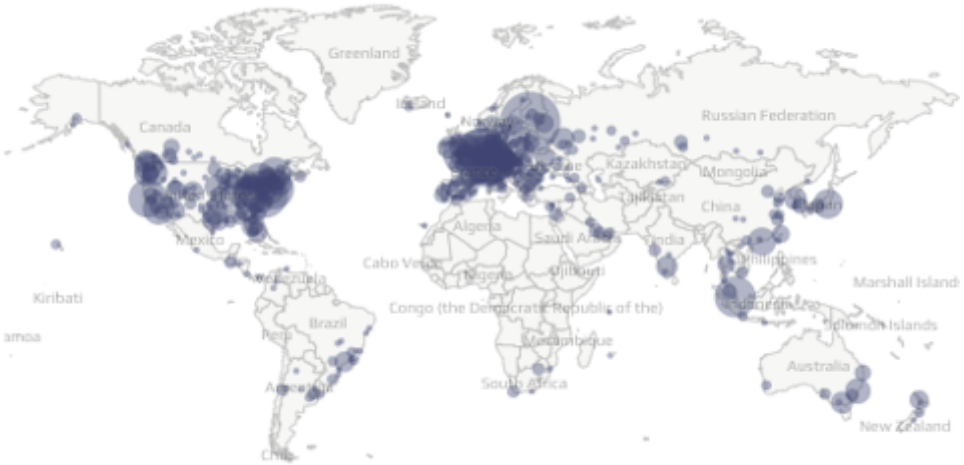


FIGURA 2.1: Concentrazione dei nodi Bitcoin in tutto il mondo

o di server, ma sono equivalenti [7], fungendo al contempo da client e da server verso gli altri nodi. La connessione è diretta, senza intermediari, grazie alla creazione di una rete Overlay in grado di mappare e di indicizzare la rete. Inoltre, i nodi, collaborano per condividere risorse e fornire servizi, scegliendo in autonomia quali risorse e quali servizi condividere. Esempi di reti P2P sono:

- **BitTorrent:** Rete per la condivisione di file, dove i dati vengono divisi in piccoli frammenti e distribuiti su più nodi della rete.
- **IPFS:** Acronimo di InterPlanetary File System, un sistema di archiviazione dati distribuito dove i file vengono archiviati su più nodi della rete.

Bitcoin è una rete di criptovaluta P2P, in cui tutti i nodi partecipano alla convalida delle transazioni e alla manutenzione della rete [8].

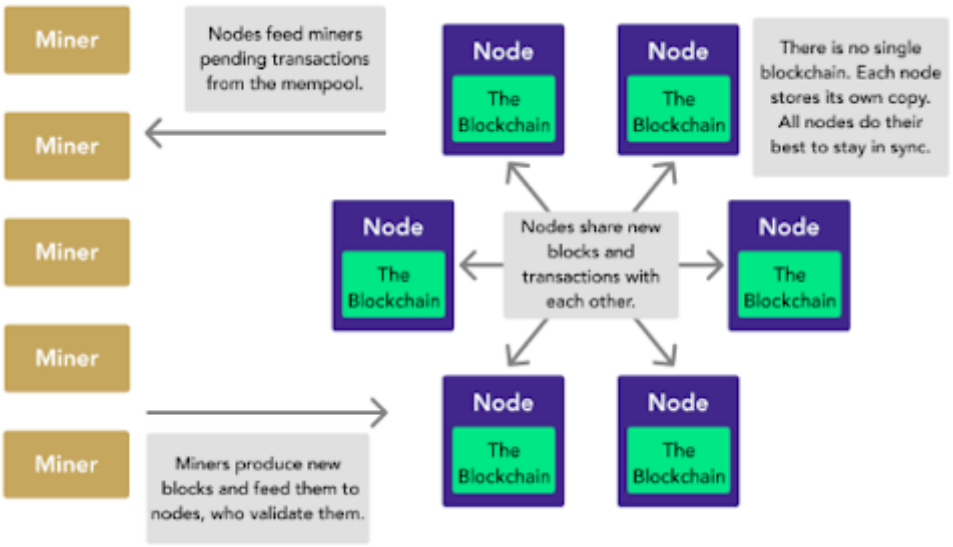


FIGURA 2.2: Componenti di rete Bitcoin

Tuttavia, i nodi che partecipano non sono tutti uguali, figura 2.2, e si differenziano dallo scopo e dai compiti che svolgono:

- **Full Node:** O nodo completo , presenta funzionalità di verifica protocollare. Mantiene l'intera blockchain e ogni nuova transazione in memoria. Se una transazione o un blocco viola il protocollo, i full node li rigettaranno. Inoltre si occupano di servizi di routing, mining e di wallet.

- **Lightweight / SPV client:** Simili ai full node, possono verificare che le transazioni siano state incluse in un blocco utilizzando una quantità di risorse inferiore rispetto ai full node poichè, conservano solo una parte e non memorizzano una copia completa della blockchain.

- **Miners:** Competono per creare nuovi blocchi implementando hardware specializzato per risolvere l'algoritmo di proof-of-work. Alcuni agiscono da nodi completi, mentre altri sono client che partecipano al mining unendo le forze per aumentare le proprie possibilità di trovare un nuovo blocco.

## 2.3 Blocchi

Ogni blocco è composto da diverse componenti, figura 2.3, e la sua dimenzione è sempre inferiore a 1 Mbyte.

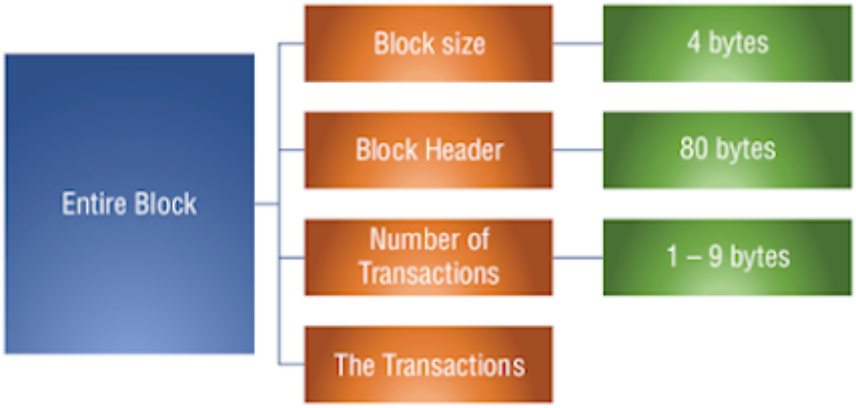


FIGURA 2.3: Costituenti di un blocco

L'intestazione, o header, si trova al di sopra delle transazioni contenute nel blocco e la sua dimenzione è di 80 bytes [9]. L'header trasporta informazioni significative, figura 2.4, che qualificano il blocco ed esprimono le sue caratterisiche:

- **Version:** Valore che esprime la versione del protocollo.
- **Previous block hash:** Hash che lo collega al blocco al precedente.
- **Merkle Root:** Meccanismo per verificare l'integrità delle transazioni all'interno di un blocco.
- **Timestamp:** Data e ora della relativa estrazione del blocco.
- **Difficulty Target:** Complessità del problema matematico risolto nella fase di mining.
- **Nonce:** Numero casuale utilizzato, durante il mining, per risolvere la proof-of-work.

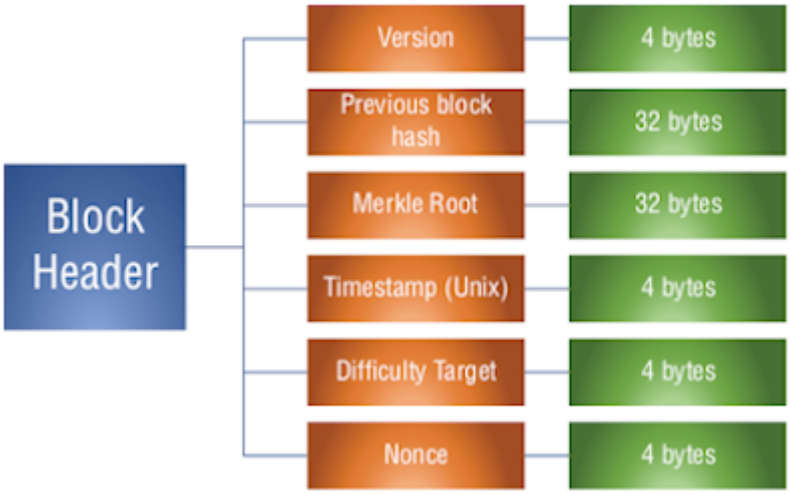


FIGURA 2.4: Costituenti header di un blocco

## 2.4 Transazioni

Le transazioni offrono una serie di caratteristiche innovative che le rendono differenti dalle transazioni monetarie tradizionali. Bonifici bancari o pagamenti con carta di credito ruotano attorno a un concetto fondamentale: il movimento di valuta da un proprietario ad un altro. In entrambi i casi, la transazione è controllata e sottoscritta da un'autorità centrale, come la Banca d'Italia. Nella Blockchain non esiste un'autorità centrale, bensì è il consenso degli utenti a fornire la validità delle transazioni.
Un indirizzo identifica univocamente il luogo in cui risiedono i bitcoin ed è il possesso della chiave privata a dimostrare la proprietà dei fondi.
Esistono tre principali tipi di transazioni Bitcoin [10]:

- **P2PKH (Pay-to-Public-Key-Hash):** Transazione standard, con un indirizzo a chiave pubblica che trasferisce valore ad un altro indirizzo. La stragrande maggioranza delle transazioni sulla blockchain Bitcoin sono P2PKH.

- **Multisignature:** Con la multifirma è necessaria più di una chiave privata per spendere il valore di un indirizzo. Questo approccio può essere utile quando ad esempio, più amministratori di una società devono approvare un pagamento.

- **P2SH (Pay-to-Script-Hash):** La multifirma è un esempio di transazione P2SH. L'indirizzo a cui viene trasferito il denaro dovrà soddisfare dei requisiti, prima che il valore possa essere nuovamente negoziato. Ad esempio, lo script di firma potrebbe richiedere più chiavi, come in una transazione multifirma, o potrebbe richiedere una password o qualsiasi altro requisito che possa essere integrato nello script.

L'mmagine in figura 2.5, mostra i due script di firma più importanti, ovvero lo script di blocco e sblocco.

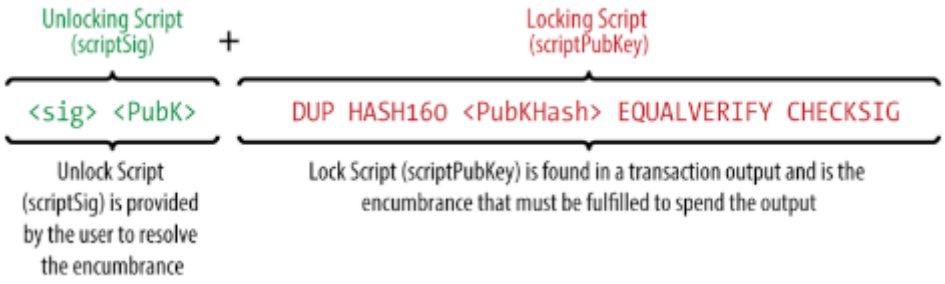


FIGURA 2.5: Esempio di script di firma

Gli script di blocco, o scriptPubKey, specificano una condizione che deve essere soddisfatta prima che i fondi possano essere spesi, definendo una serie di operazioni che devono essere portate a termine. Gli script di sblocco, o scriptSig, vengono utilizzati per soddisfare le condizioni imposte dallo script di blocco. I nodi peer della rete convalidano una transazione eseguendo contemporaneamente gli script e solo quando lo script di sblocco soddisfa le condizioni dello script di blocco, la transazione viene considerata valida.

### 2.4.1 Principio di funzionamento

Le transazioni sono degli eventi che sono costituiti, da input e da output, come mostrato in Figura 2.6.
Ogni transazione genera dei nuovi output che possono assumere due stati:

- **Speso:** se l'output generato è stato utilizzato come input di un'altra transazione.

- **Non Speso:** se l'output generato non è stato utilizzato come input di un'altra transazione.

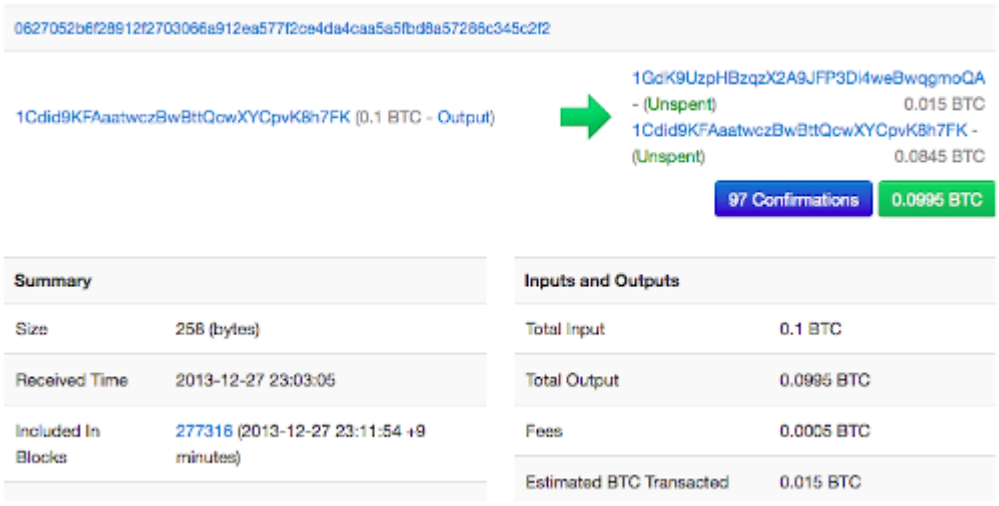


FIGURA 2.6: Esempio di una transazione

Le transazioni non spese sono anche conosciute come UTXO, acronimo di Unspent Transaction Output.
Le transazioni di Bitcoin utilizzano un UTXO già esistente come input, sbloccandolo mediante una firma digitale per generare un nuovo UTXO, che viene quindi bloccato sull'indirizzo dei nuovi proprietari.
Il metodo generalmente utilizzato per rendere complesso il monitoraggio delle transazioni consiste nell'utilizzare una chiave pubblica fresca, ovvero un nuovo "conto bancario", ogni qualvolta si scambia denaro.
Quando questo avviene, dato il numero di input e di output variabile e la pseudonimizzazione degli indirizzi, tenere traccia dei fondi e di dove vengono trasferiti diventa molto difficile.
Le nuove transazioni vengono registrate in un'area chiamata mempool, la quale rappresenta una sala di attesa dove le transazioni risiedono prima di essere incluse in un nuovo blocco. La mempool non è un luogo preciso, ma è l'insieme di tutte le informazioni che i Full node hanno sulle nuove transazioni effettuate dagli utenti.
Infatti, quando si scambia del denaro, la transazione risultante non viene inviata direttamente al destinatario ma viene propagata per tutta la rete seguendo degli algoritmi propagativi come il flooding. In questo modo si alimenta la mempool, che viene poi "alleggerita" quando, trovato nuovo blocco, le transazioni al suo interno vengono eliminate.

Ad ogni transazione, inoltre, è associata una commissione variabile ad incentivo per i miners, i quali utilizzano la propria potenza di calcolo per accrescere la catena trovando nuovi blocchi.

## 2.5 Mining

Il mining è il processo attraverso il quale avviene la creazione di nuovi bitcoin e l'estensione, in numero di blocchi, della blockchian. I miners della rete competono tra loro per risolvere la proof-of-work e il primo ad "estrarre" un nuovo blocco viene ricompensato con nuovi bitcoin e con le commissioni presenti all'interno delle transazioni. Un bitcoin o BTC è formato da unità fondamentali chiamate satoshi, in onore dell'ideatore del protocollo. Un satoshi, o Sat, rappresenta l'unità minima del bitcoin, 0.00000001 BTC, e consente la spesa di piccole somme di denaro. Il numero totale di Bitcoin estraibile è limitato a 21 milioni di unità e la ricompensa per l'estrazione di un blocco, inizialmente di 50 Bitcoin, attualmente è scesa a 6,25 Bitcoin con un dimezzamento, chiamato halving, che avviene approssimativamente ogni quattro anni o ogni 210.000 blocchi estratti [11].
Questo limite mira a controllare l'offerta di Bitcoin per garantirne la sua scarsità nel tempo e per controllarne l'inflazione, figura 2.7.

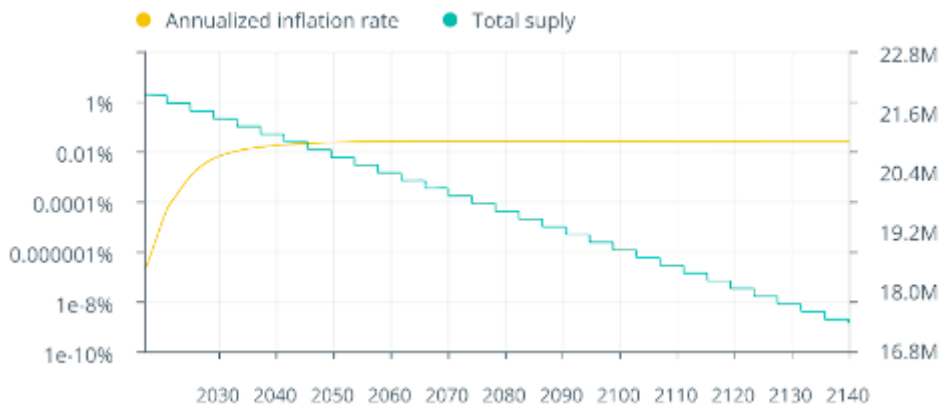


FIGURA 2.7: Fornitura futura di Bitcoin

Trovare la soluzione alla proof-of-work richiede degli sforzi computazionali notevoli che coprono appena sufficientemente l'investimento.
La maggior parte delle criptovalute richiede hardware specializzato, come ad esempio un vasto numero di schede grafiche o ASIC di fascia alta.

Le ASIC altrimenti note come Application-specific integrated circuit, sono sistemi integrati progettati per svolgere un compito specifico e nel caso di Bitcoin, il loro compito è quello di trovare degli hash molto velocemente; tuttavia questa è l'unica cosa che possono fare.
Per ridurre l'investimento, sono stati sviluppati dei pool minerari utilizzati per aumentare le probabilità di risolvere un blocco e ricevere la ricompensa in Bitcoin. L'idea di base è quella di collaborare con altri minatori in tutto il mondo, unendo le potenze di calcolo e dividendo i profitti. Esistono molti pool minerari disponibili per quasi ogni criptovaluta ed alcuni sono diventati molto potenti, figura 2.8, offrendo un'alternativa valida per chi possiede risorse limitate.

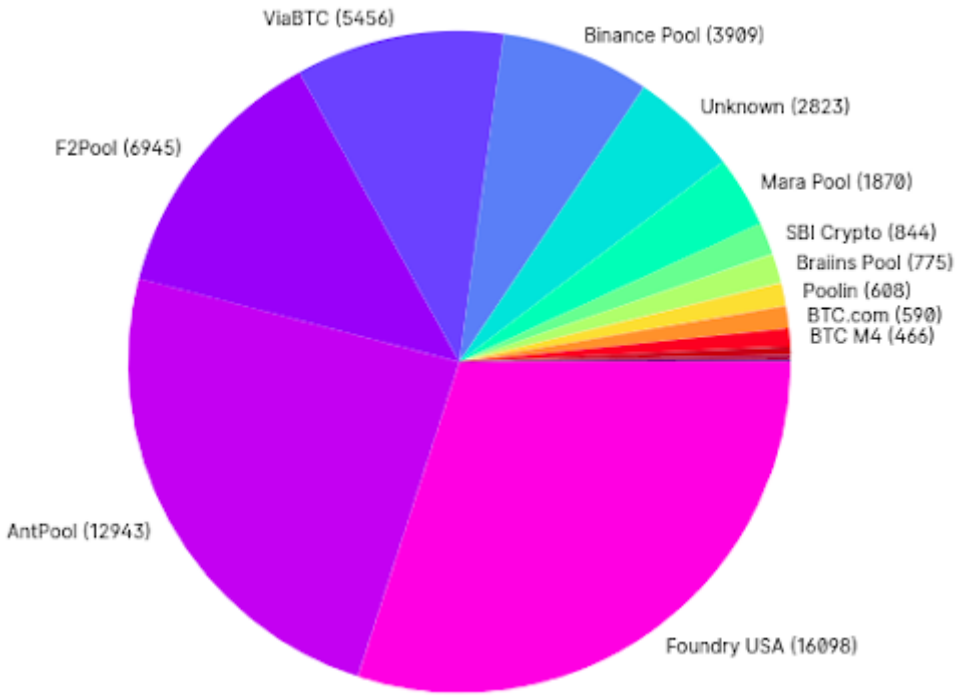


FIGURA 2.8: Ripartizione percentuale dei Bitcoin estratti dai pool minerari nell'ultimo anno.

### 2.5.1 Proof-of-Work

La proof-of-work è il concetto dietro la ricerca di una specifica soluzione ad un problema matematico. Il processo di ricerca non è dissimile dal tentativo

di forzare un lucchetto che utilizza una combinazione. Bisogna provare tutte le possibili combinazioni finché non si trova la soluzione.
In Bitcoin, i blocchi devono essere estratti ogni 10 minuti circa, non importa quanto grande diventa la potenza di calcolo che li estrae.
Per fare ciò, la rete calcola la difficulty, o difficoltà, del problema matematico in modo da garantire che trascorrano 10 minuti tra un blocco ed un altro, utilizzando una formula simile alla seguente:

$$New\,Difficulty = Old\,Difficulty * (Target\,Time/Actual\,Time)$$

La formula, calcola la nuova difficoltà in base al tempo che si desidera impiegare per estrarre 2016 blocchi (Target Time) e al tempo effettivamente trascorso per l'estrazione degli ultimi 2016 blocchi (Actual Time).
Ciò garantisce che la difficulty rimanga tale da permettere che i blocchi non siano estratti troppo velocemente o, al contrario, troppo lentamente.
Una volta estratto, il blocco viene propagato e convalidato dalla rete per garantire che non sia stato contraffatto o duplicato.
La convalida include quanto segue:

- I dati all'interno del blocco sono validi.
- Il timestamp è corretto
- Le dimensione del blocco è entro i limiti.
- La prima transazione è coinbase, cioè di generazione di bitcoin.

Ogni blocco è identificato univocamente da un hash, generato a partire dalle informazioni contenute nel blocco stesso, tra cui le transazioni, l'hash del blocco precedente e il nonce, figura 2.9.

Il nonce è un numero casuale che permette di trovare un nuovo blocco.
Per trovare un nuovo blocco i miners devono creare un blocco che abbia all'interno, l'hash del blocco precedente, le informazioni caratterizzanti l'header e delle transazioni presenti nella mempool, includendole seguendo dei criteri come la commissione associata, la dimensione della transazione o la priorità.
Il nonce, dunque, è il valore che viene cambiato di continuo per rientrare nei criteri che l'hash risultante di un nuovo blocco deve possedere. Il nuovo blocco per essere valido deve avere un codice hash che soddisfi la difficulty attuale

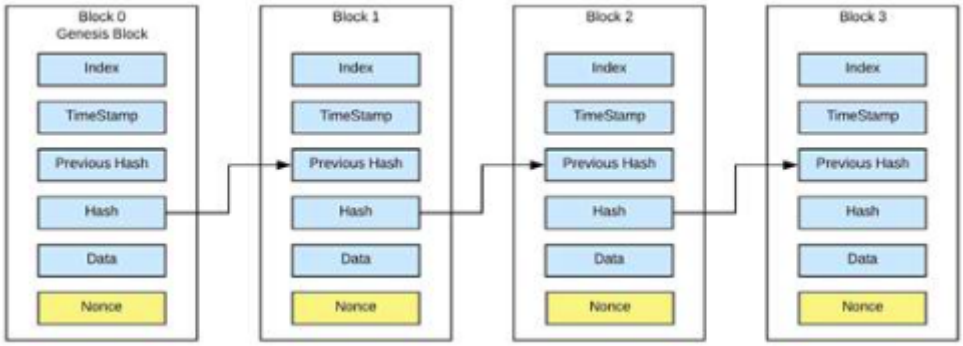


FIGURA 2.9: Connessione tra i blocchi

della rete. La difficulty determina il criterio di validità, che in genere implica la ricerca di un hash che inizia con un certo numero di zeri.

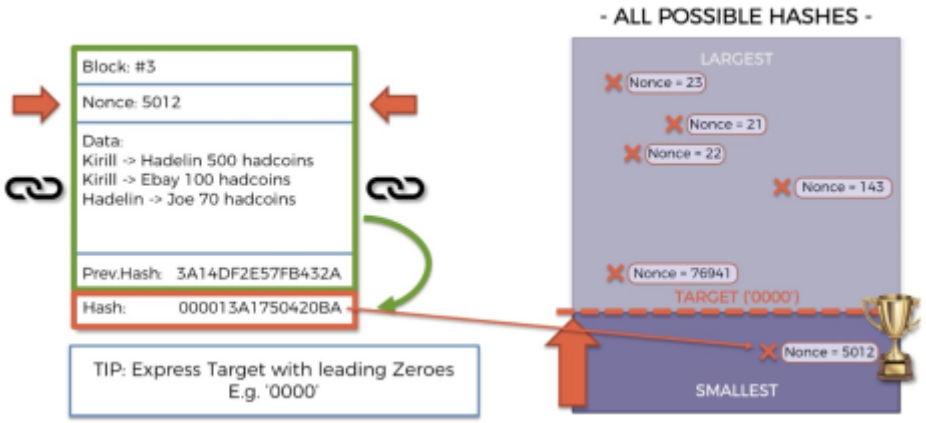


FIGURA 2.10: Ricerca di hash con quattro zero iniziali.

La figura 2.10 mostra il processo di ricerca del nonce atto a produtte un hash con un certo numero di zeri. Questo processo non è banale e viene costantemente calibrato, modificando la difficulty. Una volta trovato l'hash, i miners possono dimostrare la prova di lavoro, o proof of work, alla rete e se tutto è valido possono beneficiare dei nuovi bitcoin prodotti e delle commissioni presenti nelle transazioni. Le transazioni all'interno del blocco vengono eliminate dalla mempool, diventano parte integrante della blockchain e modificano la realtà finanziaria della rete. Una volta conclusa, la proof-of-work riparte per generare nuovi blocchi.

## 2.6 Wallet

I Wallet, o portafogli, non conservano monete ma conservano un elenco di chiavi private e chiavi pubbliche. Possono fare riferimento ad una qualsiasi transazione nella blockchian che richiede una chiave privata memorizzata. In questo modo, il wallet può generare una nuova firma digitale, facendo riferimento all'output di una transazione e all'indirizzo a cui si desidera inviare il denaro. Quindi, anche se comunemente si parla di inviare e ricevere bitcoin tramite un portafoglio, questo è solo per convenienza di linguaggio: in realtà, il portafoglio comunica con la blockchain per sbloccare e bloccare transazioni. Inoltre, un portafoglio può conservare localmente un registro delle transazioni coinvolte, l'insieme delle preferenze dell'utente e un saldo costantemente aggiornato.
Esistono due tipologie principali, figura 2.11, di wallet:

- **Hot Wallet:** sono portafogli connessi ad Internet.
- **Cold Wallet:** sono portafogli che non sono connessi ad Internet.

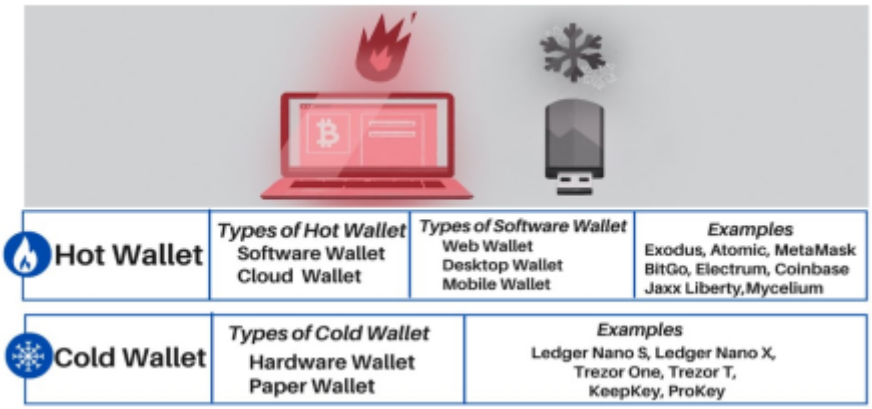


FIGURA 2.11: Hot Wallet e Cold Wallet

I software wallet possono essere configurati come hot wallet, garantendo un accesso rapido ai fondi per le transazioni quotidiane, mentre gli hardware wallet, sono spesso impiegati come cold wallet per conservare i fondi in modo sicuro, soprattutto per quantità significative di criptovalute.
Gli utenti, pertanto, devono valutare attentamente poichè ogni tipologia ha i suoi vantaggi e i suoi svantaggi in termini di sicurezza, controllo e facilità d'uso [12].

### 2.6.1 Software Wallet

Esistono diverse categorie di software wallet che si differenziano per il modo in cui gestiscono e archiviano i dati della blockchain. I tre tipi principali sono:

- **Full Node Wallet:** Questo tipo di wallet scarica l'intera blockchain e la archivia localmente sul dispositivo dell'utente. Essendo un nodo completo, partecipa attivamente alla rete Bitcoin verificando e inoltrando le transazioni. Esempi di full node wallet sono Bitcoin Core e Bitcoin Knots.

- **Thin Client Wallet (o SPV Wallet):** Questo tipo di wallet non scarica l'intera blockchain, ma solo le intestazioni dei blocchi (headers), riducendo così lo spazio di archiviazione necessario. Gli SPV (Simplified Payment Verification) wallet verificano la validità delle transazioni senza dover memorizzare l'intera blockchain. Esempi di thin client wallet includono Electrum e MultiBit.

- **Online Wallet (o Web Wallet):** Questi sono wallet che memorizzano le chiavi private degli utenti su server online gestiti da terze parti. Gli utenti possono accedere ai loro fondi tramite un'interfaccia web, senza dover scaricare alcun software sul proprio dispositivo. Mentre offrono convenienza, gli online wallet possono essere considerati meno sicuri rispetto ai wallet che conservano le chiavi private localmente. Esempi di online wallet sono Coinbase e Blockchain.info.

### 2.6.2 Hardware Wallet

Gli hardware wallet sono dispositivi fisici progettati per fornire un'opzione altamente sicura per conservare e proteggere le chiavi private. Alcuni esempi sono il Ledger NanoS, il Trezor Wallet e Keepkey.
Gli hardware wallet spesso seguono lo standard BIP39, o Bitcoin Improvement Proposal 39, che descrive un metodo per generare una sequenza di parole mnemoniche (seed phrase) che possono essere utilizzate per derivare un set di chiavi private. Le seed phrase, note anche come frasi di recupero o frasi di backup, sono un insieme di parole che contengono tutte le informazioni necessarie per il ripristino delle chiavi private [13]. Sono ampiamente utilizzate, anche da molti portafogli software, poichè consentono la memorizzazione del

segreto in modo sicuro ed affidabile, riducendo il rischio di perdita, guasto o furto.

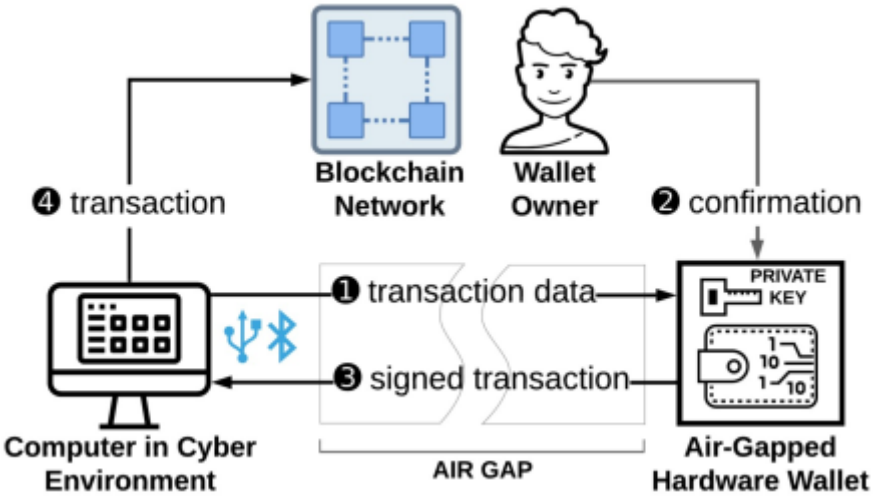


FIGURA 2.12: Utilizzo di un hardware wallet

Inizializzata la chiave privata, essa rimane criptata in un chip Secure Element separato ed offline, figura 2.12
Un'altro standard molto seguito è il BIP32, acronimo di Bitcoin Improvement Proposal 32, che descrive come generare una gerarchia deterministica di chiavi [14]. Ciò consente di organizzare e di generare un numero illimitato di coppie di chiavi per l'invio e la ricezione di criptovalute.

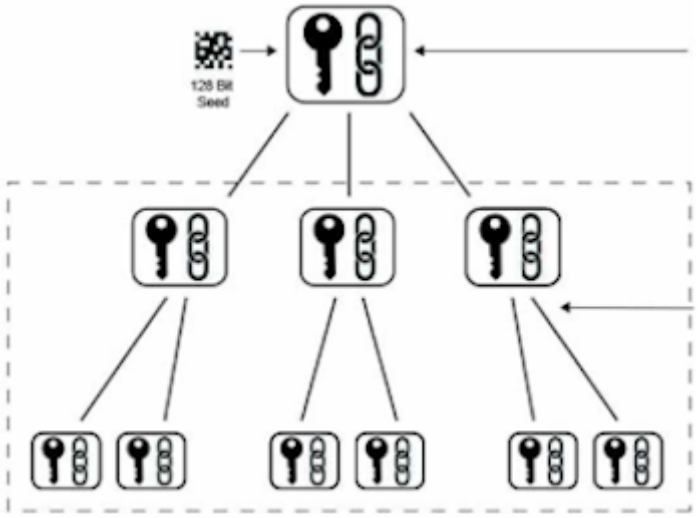


FIGURA 2.13: Gerarchia deterministica di chiavi BIP32

# Capitolo 3

# Analisi on Chain

L'analisi on chain, o blockchain analysis, racchiude i processi di aggregazione, analisi e controllo dei dati delle blockchain per monitorare la provenienza delle transazioni, i movimenti dei flussi monetari e gli schemi di comportamento degli utenti, al fine di costruire modelli visivi e rappresentativi delle informazioni. I dati ottenuti da queste tecniche possono essere utilizzati per molteplici scopi, come la lotta al crimine informatico o il monitoraggio delle tendenze di mercato. L'analisi forense di criptovalute viene spesso svolta da società private o governative, come Chainalysis ed Elliptic. Queste società offrono dei servizi che permettono di identificare periodicamente un numero molto elevato di indirizzi che sono coinvolti in una moltitudine di attività criminali.
Le transazioni di asset crittografico sono intrinsecamente pseudoanonime, pertanto queste società, oltre ad alti servizi, collaborano con le forze dell'ordine, cercando di de-anonimizzare le transazioni, per consentire loro di identificare l'individuo o l'organizzazione criminale dientro gli indirizzi delle blockchain.

## 3.1 Casistica dei crimini informatici

Ogni giorno, pirati informatici e hacker escogitano nuove strategie per sfruttare le peculiarità delle criptovalute al fine di commettere ogni tipo di reato e

sfuggire alla giustizia. Esistono cinque macro categorie che descrivono i crimini legati alle blockchain:

- **Compravendita di merce illegale:** Le criptovalute vengono utilizzate da criminali informatici e non per acquistare o vendere merce illecita o illegale. La compravendita avviene principalmente sul dark web, una parte di internet non indicizzata dai motori di ricerca e accessibile solo tramite software specifico che maschera l'indirizzo IP. I venditori pubblicano i prodotti sul mercato nero del dark web, dove gli acquirenti possono acquistarli in modo anonimo, sfruttando le criptovalute.

- **Furti di criptovalute:** I furti sono un problema crescente, con miliardi di dollari rubati ogni anno. I criminali informatici utilizzano una miriade di vettori di attacco per rubare la criptovaluta, come il phishing, le truffe di investimento, l'hacking di exchange ed il social engineering. Un esempio emblematico è il furto ai danni di Binance, una piattaforma di trading e di exchange, avvenuto nel 2019. Una vulnerabilità nel sistema di sicurezza ha permesso agli hacker di accedere agli hot wallet dell'exchange e di trasferire su un wallet esterno 7.000 Bitcoin.

- **Estorsioni:** Le estorsioni che impiegano le criptovalute sono un tipo di crimine informatico in cui gli hacker minacciano di pubblicare dati personali o sensibili, o minacciano di danneggiare i sistemi informatici minandone le funzionalità. Il punto chiave è la richiesta di un riscatto, che se non pagato, porta alla pubblicazione dei dati o al danneggiamento dei sistemi informatici.

- **Riciclaggio di denaro:** Il riciclaggio di denaro è il processo atto a nascondere l'origine illegale dei proventi di reato per integrarli nell'economia legale. Su una blockchain, il denaro può essere riciclato spostando monete attraverso mixer o exchange ed acquistando criptovalute con denaro illegale per poi trasferirle su diversi wallet anonimi così da convertire la criptovaluta in valuta fiat. Inotre, possono essere utilizzate per evadere le tasse. I criminali possono nascondere, in diversi modi, i loro guadagni acquistando criptovalute e utilizzandole successivamente per acquistare beni e servizi.

Un esempio tristemente famoso di sfruttamento non etico delle criptovalute sono gli attacchi ransomware. Il ransomware è un malware cioè un software


FIGURA 3.1: Infezione da ransomware WannaCry

dannoso, progettato per compromettere o sfruttare qualsiasi tipo di dispositivo, servizio o rete programmabile [15].

In particolare, il ransomware, è un tipo specifico di malware il cui compito principale è quello di limitare o impedire l'accesso ai dati o ai dispositivi. Per raggiungere lo scopo il ransomware utilizza la crittografia, rendendo i file illeggibili ed inutilizzabili. Per ripristinare l'accesso ai dati, gli hacker sono soliti chiedere un riscatto, spesso in criptovaluta, figura 3.1.

Per tutti i motivi sopra elencati, monitorare l'utilizzo non etico delle criptovalute da parte degli utenti della rete è un obiettivo primario nella lotta al cybercrimine. Seguire i flussi di denaro può essere molto complesso poichè i criminali agiscono sempre cercando di nascondere le proprie tracce. Le tecniche di analisi on chain sono dunque di fondamentale importanza per scoprire le attività criminali dietro gli scambi di denaro e assicurare i responsabili di eventuali crimini alla giustizia.

## 3.2 Tecniche di analisi

Costruire un profilo indentitario di una persona o di un gruppo partendo da un indirizzo appartenente ad una blockchain richiede la raccolta di molte informazioni e lo svolgersi di molte indagini su:

- **Cronologia e tipologia:** Classificazione della tipologia e origine delle transazioni basandosi sul riconoscimento della provenienza e sulla natura delle transazioni associate ad un indirizzo, come scambi su exchange, donazioni o attività di mining.
- **Analisi temporale:** Studio delle transazioni nel tempo basandosi sui timestamp per identificare pattern di comportamento o eventi inusuali.
- **Estrazione di micromessaggi:** Ricerca di informazioni codificate all'interno delle transazioni, ad esempio, negli script di blocco e sblocco.
- **Identificazione di cluster:** Applicazione di algoritmi di clustering per identificare relazioni e connessioni su una moltitudine di indirizzi al fine di ricondurli ad un unico proprietario o gruppo.
- **Analisi del network:** Valutazione del flusso di transazioni associato all'indirizzo in esame attraverso l'analisi delle sue interazioni con altri indirizzi.
- **Informazioni off-chain:** Integrazione di dati provenienti da fonti pubbliche, come social media, forum online e database di intelligence tramite strumenti di OSINT (Open Source Intelligence).

Le tecniche utilizzate sono numerose ed eterogenee poichè utilizzano una notevole varietà di strumenti. Le analisi on-chain e off-chain forniscono informazioni preziose su attività sospette, minacce e salute della criptovaluta.
Il loro utilizzo individuale o in combinazione dipende da diversi fattori come lo scopo dell'analisi, le risorse disponibili ed il livello di rischio, tuttavia l'utilizzo di diverse tecniche in combinazione può fornire una visione più completa e olistica della sicurezza di una blockchain.

## 3.3 Transaction Graph Analysis

La Transaction Graph Analysis, o analisi del grafo delle transazioni, è una tecnica che prende in esame la struttura delle transazioni costruendone un grafo visuale. Le entita coinvolte vengono rappresentate come dei nodi e gli stessi vengono interconnessi tra di loro grazie a delle relazioni.
Il grafo risultante viene sottoposto ad analisi visuale o computazionale per scoprire eventuali modelli, schemi o connessioni che possono favorire le indagini forensi. La possibilità di visualizzare il grafo assume un ruolo fondamentale nell'analisi, poiché consente agli investigatori di ottenere una migliore comprensione della rete e delle transazioni [16].
Analizzando la struttura e le proprietà del grafo è possibile ottenere informazioni preziose su diversi aspetti della blockchain, tra cui:

- **Flusso delle transazioni:** Consente di tracciare il percorso che le transazioni seguono tra diversi indirizzi e di identificare le relazioni tra le controparti coinvolte.
- **Attività sospette:** Individuazione di anomalie nel comportamento di transazioni o nodi, come picchi di volume o transazioni ricorrenti verso lo stesso indirizzo.
- **Identificazione di cluster:** Possibile scoperta di cluster di indirizzi che presentano un elevato numero di transazioni reciproche, suggerendo una possibile comune proprietà o affiliazione.

L'analisi tramite grafi visuali consente la creazione di una visione olistica delle transazioni e delle entità coinvolte, collegandole in un unico contesto. Permette agli investigatori di ottenere una migliore comprensione della rete, di identificare le relazioni tra le entità coinvolte e di scoprire modelli di comportamento anomalo. La scoperta di nuovi connessioni nascoste può evidenziare attività illecite che potrebbero non essere evidenti da un'analisi individuale delle transazioni. Lo strumento sviluppato si ispira a questa filosofia di pensiero.

# Capitolo 4

# Glockchain: uno sguardo allo strumento

## 4.1 Architettura del software

L'idea che ha portato allo sviluppo di Glockchain è nata dalla necessità di avere strumenti Open Source e gratuiti per l'Analisi on-Chain. La scelta ricaduta su Python e Neo4j è da attribuire alle caratteristiche intrinseche delle tecnologie e alle loro peculiarità. Il software cosi ideato è composto da due parti:

- GlockChain
- Neo4j

La componente Glockchain è responsabile del processo ETL (estrazione, transformazione,caricamento) del dato. Mentre la componente Neo4j si occupa della memorizzazione, organizzazione e visualizzazione del dato.

## 4.2 Neo4j

Neo4j è un database a grafo open source, leader nel suo campo, utilizzato per modellare, memorizzare e interrogare dati connessi. A differenza dei database

relazionali tradizionali, che si basano su tabelle e schemi rigidi, Neo4j utilizza un modello di dati flessibile basato su grafi, che rappresenta le relazioni tra entità come nodi e relazioni [17].

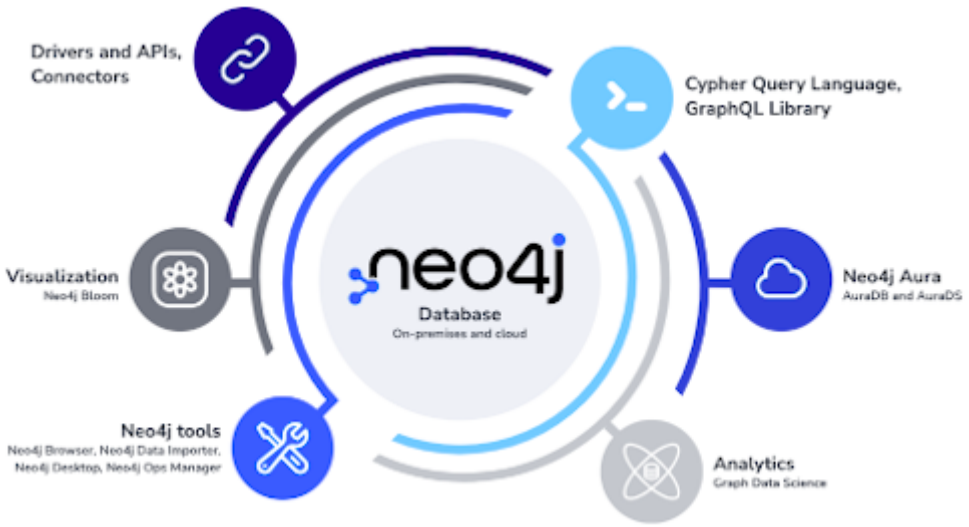


FIGURA 4.1: Panoramica dell'ecosistema Neo4j

Neo4j mette a disposizione diverse funzionalità, come mostra la figura 4.1, ed è utilizzato per studi in diversi campi applicativi:

- **Social Network Analysis:** Analisi delle relazioni tra utenti di social network per identificare di influencer e trend.
- **Ricerca di frodi:** Identificazione di transazioni fraudolente tramite l'analisi di modelli di comportamento nei dati finanziari.
- **Raccomandazione di prodotti:** Personalizzazione dei prodotti raccomandati agli utenti, in base alla loro cronologia di acquisto.

La forza della piattaforma risiende nel potere di memorizzare e rappresentare qualsiasi tipo di dato utilizzando alcuni concetti di base:

- **Nodi:** rappresentano le entità di un dominio.
- **Etichette:** modellano il dominio, raggruppando i nodi in insiemi.
- **Relazioni:** collegano i nodi.

- **Proprietà:** valori aggiuntivi che qualificano i nodi e le relazioni

Come illustrato nella figura 4.2, i dati in Neo4j vengono memorizzati sotto forma di nodi. I nodi appartengono a degli insiemi distinti, sono qualificati da proprietà e sono collegati tra di loro tramite relazioni direzionali. Le relazioni a loro volta possono essere caratterizzate da proprietà che ne arricchiscono il significato.

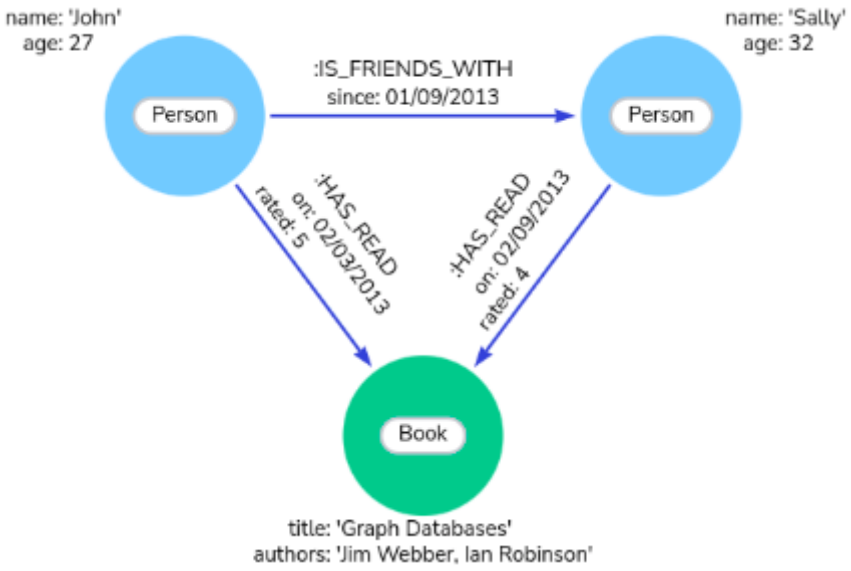


FIGURA 4.2: Modello a grafo utilizzato da Neo4j

L'esempio mostra tre nodi appartenenti a due insiemi, identificati dalle etichette "Person" e "Book".
I nodi "Person" presentano le proprietà "name" ed "age" mentre il nodo "Book" presenta le proprietà "title" e "authors". I nodi sono connessi da relazioni che permettono di intuire, dal nome e dalle proprietà, la connessione che intercorre tra un nodo e i suoi "corrispettivi".
Neo4j per lavorare con grafi e consentire agli utenti di eseguire operazioni come l'inserimento, l'aggiornamento, l'eliminazione e l'interrogazione dei dati utilizza un linguaggio, chiamato Cypher, progettato specificatamente per questo compito. È un linguaggio intuitivo ed espressivo, che somiglia leggermente ad SQL (Structured Query Language) per la sintassi dichiarativa, tuttavia se ne distacca significativamente in termini di filosofia, struttura dei dati e sintassi.
La figura 4.3 evidenzia il codice Cypher necessario per la creazione del grafo in figura 4.2

```
MERGE (j:Person {name: 'John'})
  ON CREATE set j.age = 27
MERGE (s:Person {name: 'Sally'})
  ON CREATE set s.age = 32
MERGE (b:Book {title: 'Graph Databases'})
  ON CREATE set b.authors = ['Jim Webber', 'Ian Robinson']
MERGE (j)-[rel1:IS_FRIENDS_WITH]->(s)
  ON CREATE SET rel1.since = '01/09/2013'
MERGE (j)-[rel2:HAS_READ]->(b)
  ON CREATE SET rel2.on = '02/03/2013', rel2.rated = 5
MERGE (s)-[rel3:HAS_READ]->(b)
  ON CREATE SET rel3.on = '02/09/2013', rel3.rated = 4
```

FIGURA 4.3: Esempio di codice Cypher

Inserito il codice nella dashboard di Neo4j è possibile visualizzare il risultato della query, figura 4.4, e notare come vengono mostrate, per ogni nodo, le proprietà.
La dashboard Neo4j Browser è un'interfaccia grafica interattiva che consente agli utenti di relazionarsi con il database Neo4j utilizzando Cypher. Mette a disposizione degli strumenti per l'editor delle query, per l'esplorazione del grafo e per la visualizzazione dei risultati. Tuttavia per grafi molto complessi o di elevate dimenzioni, in termini di numero di node, è consigliato l'utilizzo dello strumento Neo4j Bloom che verrà mostrato nello studio del caso d'uso.

## 4.3 Glockchain

Glockchain è la componente operativa che si interfaccia con l'utente e si occupa di estrarre, trasformare e caricare i dati nel database. Realizzata completamente in linguaggio Python, è composta da tre moduli:

- **GlockChain.py**
- **Bitcoin_interactor.py**
- **Neo4j_Connector.py**

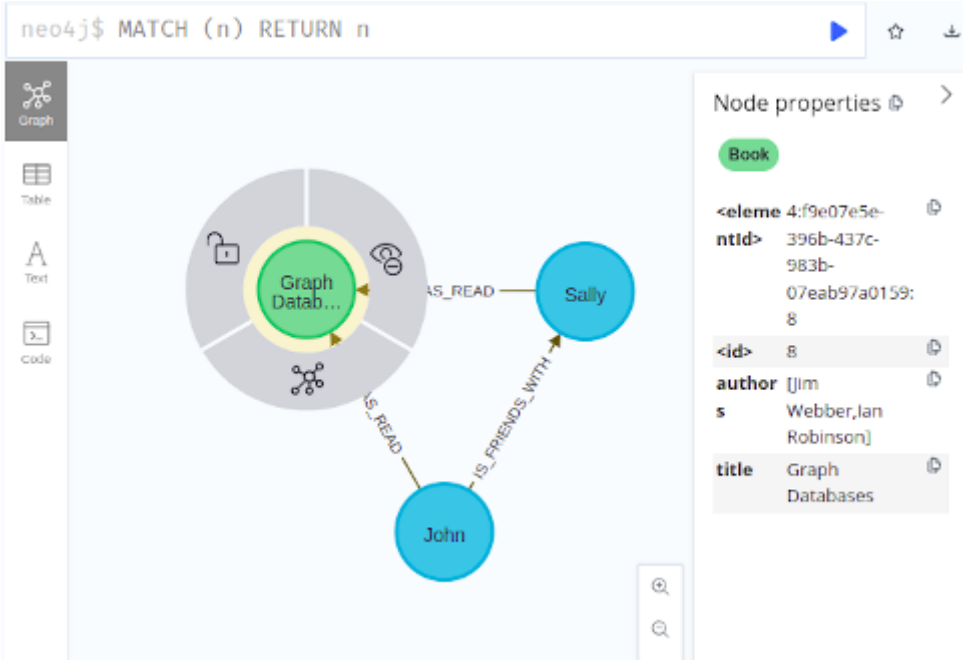


FIGURA 4.4: Dashboard Neo4j Browser

### 4.3.1 Glockchain.py

Questo file contiene le funzionalità che permettono all'utente di interagire con il sistema. Il programma permette di gestire le informazioni su Transazioni, UTXO ed in particolare, l'utente può memorizzare o eliminare tali dati dal database inserendone l'hash. Il frammento di codice mostrato in 4.1 illustra la funzione store_Database() la quale, su richiesta dell'utente, si occupa di espandere la conoscenza su nuove Transazioni.

```
def store_Database():
 interactor = Bitcoin_interactor()
 connector = Neo4j_connector(url,auth)
 while True:
    print("-----------------------")
    print("What you want to store?")
    print("1. Store Transaction")
    print("2. Exit")
    try:
        choice = int(input("Choice your action-> "))
    except ValueError:
        print("Invalid input. Please enter a number.")
        continue
```

```
    if choice == 1:
        hash = input("\nInsert the TXs to store (hash comma
    separated) -->")
        hash_list = hash.split(',')
        for item in hash_list:
         interactor.bitcoin_transaction_DataRetrieval(item)
         connector.store_transaction_Info(interactor)
    elif choice == 2:
        print("Exiting store functionalities.")
        print("------------------------")
        break
    else:
        print("Invalid choice. Please enter a number between 1
     and 2.")
```

LISTING 4.1: store_Database()

La funzione istanzia due oggetti, *interactor* e *connector*, appartenenti alle classi definite nei moduli *Bitcoin_interactor.py* e *Neo4j_Connector.py*. L'utilizzo in sequenza dei suddetti oggetti è necessario poichè, *interactor* contiene i metodi necessari per la raccolta e la trasformazione delle informazioni, mentre *connector* contiene i metodi necessari per interagire con Neo4j.

### 4.3.2 Bitcoin_interactor.py

Il modulo mostrato in 4.2 contiene la classe le quali funzionalità permettono di interagire con la blockchain.
Il costruttore inizializza diversi dizionari che conterranno le informazioni recuperate dal metodo *bitcoin_transaction_DataRetrieval(self,hash)*.

```
import requests
from datetime import datetime

class Bitcoin_interactor:

    def __init__(self):
        self.tx_json = {} # json of the entire request
        self.tx_info = {} # transantion info
        self.utxo_inputs = {} # Input UTXO
        self.utxo_outputs = {} # Output UTXO
```

```
    # Method that request Tx info
    def bitcoin_transaction_DataRetrieval(self,hash):
        url = f'https://api.blockchair.com/bitcoin/dashboards/
transaction/{hash}'
        response = requests.get(url)
        if response.status_code == 200:
            self.tx_json = response.json()
            self.UTXO_extractor(hash)
        else:
            print(f"Request failed with status code {response.
status_code}, please controll the hash")

    # Method to extract the TX info
    def UTXO_extractor(self,hash):
        data = self.tx_json['data']
        tx = data[hash] # Selecting the right keys

        self.tx_info = tx["transaction"] # The info of the tx
        self.utxo_inputs  = tx['inputs'] # The input Utxo
        self.utxo_outputs = tx['outputs'] # The output Utxo
```

LISTING 4.2: bitcoin_transaction_DataRetrieval(self hash)

Suddetto metodo effettua una richiesta al provider di servizi Blockchair.com il quale espone delle API, o Application Programming Interface, a favore di chiunque volesse interagire con la blockchain. In questo caso la richiesta è effettuata su un endpoint che permette di recuperare le informazioni su una Transazione. La risposta, in formato Json, contiene tutti i dati necessari affinche venga schematizzata la transazione all'interno del database.
Il metodo *UTXO_extractor(self,hash)* estrae le informazioni di interesse, che riguardano la transazione e gli input ed output, e le memorizza sui dizionari. I dati, in questo modo organizzati, sono pronti per essere inseriti nel database.

### 4.3.3 Neo4j_Connector.py

All'interno di questo modulo sono presenti i metodi per la creazione o eliminazione dei nodi all'interno di Neo4j. Riempiti i dizionari grazie al metodo *interactor.bitcoin_transaction_DataRetrieval(self,hash)*, l'oggetto *interactor* verrà passato come argomento del metodo *connector.store_transaction_Info(self , interactor)* che contiene le chiamate ai metodi necessari per la creazione dei nodi Transaction ed UTXO.

```
from neo4j import GraphDatabase

class Neo4j_connector:

    def __init__(self,url,auth):
        self.url = url
        self.auth = auth

    def store_transaction_Info(self, interactor):
        try:
            with GraphDatabase.driver(self.url, auth=self.auth
    ) as driver:
                driver.verify_connectivity()
                print("Processing storage:")
                self.create_TX(driver, interactor.tx_info)
                self.create_input_UTXO(driver, interactor.
    utxo_inputs)
                self.create_output_UTXO(driver, interactor.
    utxo_outputs)

                print("Storage Complete")
        except Exception as e:
            print(f"The method store_transaction_Info has
    launched an Exception: {e}")

    def create_TX(self,driver,tx_info):
        summary = driver.execute_query(
        """
        MERGE (:Transaction {
            name: $name,
            time: $time,
            block_id: $block_id,
            coinbase: $coinbase,
            input_count: $input_count,
            output_count: $output_count,
            input_value: $input_value,
            input_value_usd: $input_value_usd,
            output_value: $output_value,
            output_value_usd: $output_value_usd
        })
        """,
        name =tx_info['hash'],
```

```
        time=tx_info['time'],
        block_id=tx_info['block_id'],
        coinbase=tx_info['is_coinbase'],
        input_count=tx_info['input_count'],
        output_count=tx_info['output_count'],
        input_value=tx_info['input_total'],
        input_value_usd=tx_info['input_total_usd'],
        output_value=tx_info['output_total'],
        output_value_usd=tx_info['output_total_usd']
    ).summary
        print("TX Stored --> {}" .format(tx_info['hash']))

    #Method to create an UTXO node from the input of the TX
    def create_input_UTXO(self,driver,utxo_inputs):

        for item in utxo_inputs:
            summary = driver.execute_query(
            """
            MERGE (:UTXO {
                name: $name,
                time: $time,
                block_id: $block_id,
                recipient: $recipient,
                value: $value,
                value_usd: $value_usd,
                is_spent: $is_spent,
                spending_transaction_hash:
    $spending_transaction_hash
            })
            """,
            name =item['transaction_hash'] +":"+ str(item['
    index']),
            time=item['time'],
            block_id=item['block_id'],
            recipient=item['recipient'],
            value=item['value'],
            value_usd=item['value_usd'],
            is_spent=item['is_spent'],
            spending_transaction_hash=f"{item['
    spending_transaction_hash']}"

        ).summary
```

```
            print("UTXO--> {}:{}" .format(item['
    transaction_hash'], item['index']))

            self.create_input_Relation(driver, item['
    transaction_hash'] +":"+ str(item['index']), item['
    spending_transaction_hash'])


    def create_input_Relation(self,driver,utxo_id,tx_id):
        records, summary, keys = driver.execute_query(
        f"""
            MATCH (u:UTXO) WHERE u.name='{utxo_id}'
            MATCH (t:Transaction) WHERE t.name='{tx_id}'
            MERGE (u)-[:INPUT]->(t)
        """
        )
```

LISTING 4.3: Neo4j_connector.py

Per motivi di compattezza il frammento di codice mostra i metodi utilizzati per la creazione del nodo Transaction, del nodo UTXO di input e il metodo che permette la creazione della loro relazione. Ovviamente sono presenti anche i metodi per la creazione dei nodi e relazioni con i nodi UTXO di output e i metodi per l'eliminazione delle informazioni [18]. Le loro funzionalità sono:

- **store_transaction_Info(self,interactor):** Metodo chiamato dal modulo Glockchain.py, inizializza l'oggetto driver necessario per la connettività al database e lancia i metodi per la creazione dei nodi.
- **create_TX(self,driver,tx_info):** Metodo che crea un nodo assegnando l'etichetta Transaction, associa le proprietà salienti della transazione e grazie alla parola chiave *MERGE* evita eventuali duplicati.
- **create_input_UTXO(self,driver,utxo_inputs):** Metodo che crea un nodo assegnando l'etichetta UTXO, associa le proprietà salienti e lancia il metodo che permette la creazione della relazione.
- **create_input_Relation(self,driver,utxo_id,tx_id)**: Metodo che crea la relazione tra l'UTXO di input e la Transazione.

I metodi sfruttano la libreria di python per interagire con l'istanza di Neo4j la quale permette di effettuare le query necessarie per lo scopo.

## 4.4 Funzionamento

Avviata l'applicazione è possibile inserire o eliminare transazioni multiple separandole da una virgola.

```
What you want to store?
1. Store Transaction
2. Exit
Choice your action-> 1
Insert the TXs to store (hash comma separated)
-->eb6e27cf227ecd7a7859cea5949a5d19a31228f701785a7f92ad88cbb7e3d721,5ede8d8
51c232ad9c174c2b0a378b4d9af3449129f67a2a1ac49d83be41c107b
Processing storage:
TX--> eb6e27cf227ecd7a7859cea5949a5d19a31228f701785a7f92ad88cbb7e3d721
UTXO--> 7a40ed9e472f984b08170b13af0665e43811ed634a1408942252d212ff3fa566:99
UTXO--> 66abb16c128624ac79b04b8652a97c0b284cf71cdba9b9c88e5c01e2eadf2e77:1
UTXO--> eb6e27cf227ecd7a7859cea5949a5d19a31228f701785a7f92ad88cbb7e3d721:0
UTXO--> eb6e27cf227ecd7a7859cea5949a5d19a31228f701785a7f92ad88cbb7e3d721:1
Storage Complete
Processing storage:
TX--> 5ede8d851c232ad9c174c2b0a378b4d9af3449129f67a2a1ac49d83be41c107b
UTXO--> eb6e27cf227ecd7a7859cea5949a5d19a31228f701785a7f92ad88cbb7e3d721:0
UTXO--> eb6e27cf227ecd7a7859cea5949a5d19a31228f701785a7f92ad88cbb7e3d721:1
UTXO--> 5ede8d851c232ad9c174c2b0a378b4d9af3449129f67a2a1ac49d83be41c107b:0
UTXO--> 5ede8d851c232ad9c174c2b0a378b4d9af3449129f67a2a1ac49d83be41c107b:1
Storage Complete
----------------------
```

FIGURA 4.5: Caricamento Transazioni ed UTXO

Per inserire delle nuove transazioni, sarà sufficiente immetterne l'hash ed attendere la conclusione del processo.
La figura 4.5 mostra l'inserimento di due transazioni dove si deduce la loro composizione, formata da quattro UTXO ciascuna.
Il risultato, mostrato in figura 4.6, evidenzia come le transazioni, in rosso, sono connesse tra di loro grazie alle relazioni che hanno con i nodi UTXO di input e di output, in blu. Infatti, le due transazioni apparentemente disconnesse in realtà condividono degli UTXO che vengono creati nella prima transazione per poi essere spesi nella seconda, creando altri due UTXO.
In questo modo è possibile seguire il flusso delle transazioni lungo la catena, creandone una connessione grazie a UTXO spesi e a UTXO creati. Il processo inoltre, crea in automatico delle nuove relazioni, se presenti, con i nodi preesistenti nel database. Infatti, le query sono progettate per scoprire relazioni ed aggiungerle al grafo grazie alle proprietà *spending_transaction_hash* e *name* degli UTXO. Se esiste una connessione tra queste proprietà e la proprietà identificativa, cioè l'hash di un nodo Transaction, allora le nuove componenti verranno connesse tra di loro, illuminando nuove connessioni.

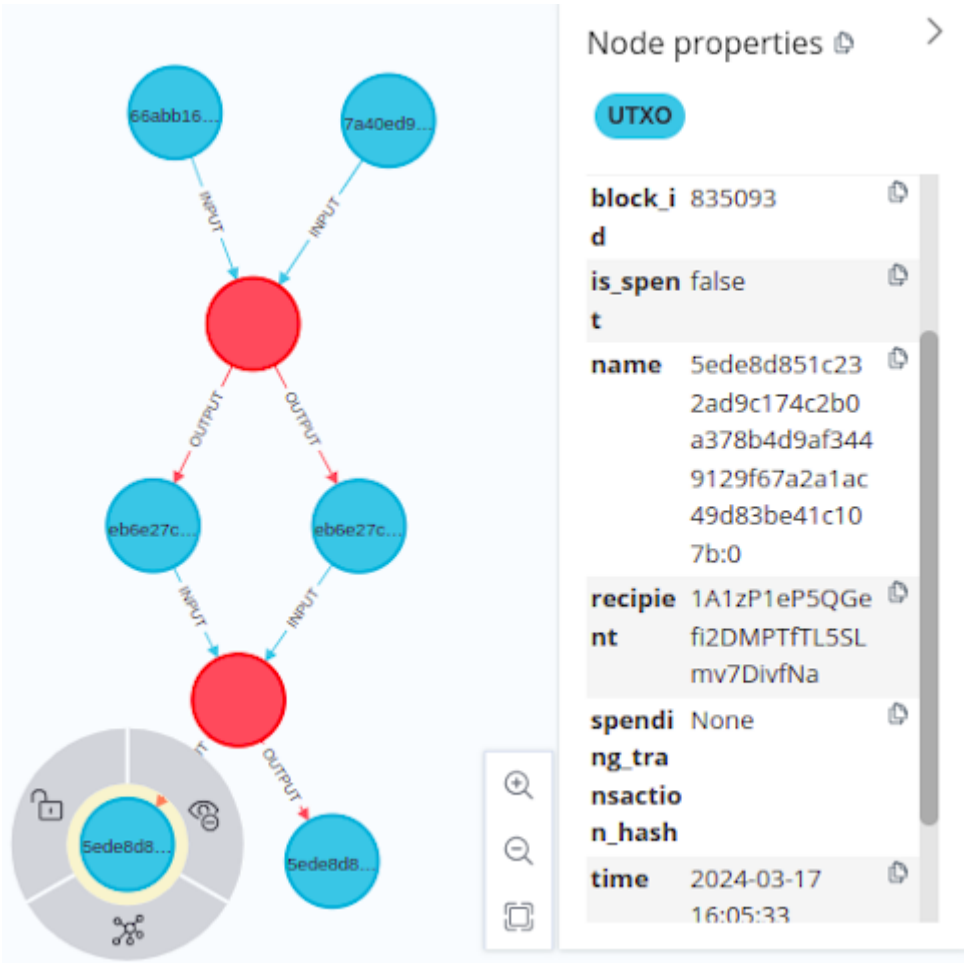


FIGURA 4.6: Grafo di esempio

Infine, per espandere il grafo è sufficiente prendere le proprietà *name* o *spending_transaction_hash* dei nodi UTXO, inserirli su Glockchian e attendere il processamento dei dati. Una o più nuove transazioni, con i relativi UTXO, verrànno memorizzate andando ad arricchire di nuovi nodi e relazioni il database.

# Capitolo 5

# Furto di Bitcoin

Le potenzialità dell'applicativo sono state testate su un caso di furto reale. Il software è stato impiegato per seguire il flusso dei Bitcoin rubati lungo la catena delle transazioni successive al furto, in modo da costruire e visualizzare il grafo delle transazioni.
L'indagine ha preso il via dal riconoscimento delle transazioni sospette, per giungere infine al riconoscimento di pattern ciclici. Tramite l'analisi visiva è stato possibile individuare dei modelli di comportamento rispetto alle modalità in cui il denaro è stato ramificato. L'indagine evidenzia schemi ricorrenti che supportano l'ipotesi di smurfing dei Bitcoin, una tecnica che include l'invio di piccole somme di denaro a vari indirizzi per riciclare la refurtiva e renderla resiliente al tracciamento.
Il capitolo si chiude con una discussione sulle possibili traiettorie di integrazione e sviluppo del software con altre metodologie e tecnologie.

## 5.1 Indagine investigativa

Lo sviluppatore di Bitcoin Core, software che implementa il protocollo Bitcoin originale, Luke Dashjr il primo gennaio 2023 ha affermato, sulla piattaforma X, che il suo wallet è stato violato a causa di una compromissione della chiave PGP, o Pretty Good Privacy, un sistema crittografico utilizzato per proteggere

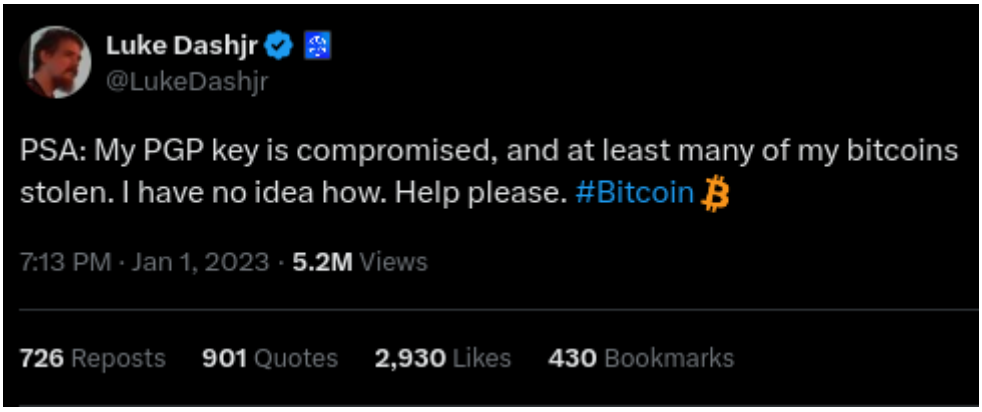


FIGURA 5.1: Post di Luke Dashjr post su X

le informazioni online, figura 5.1. Il portafoglio di Dashjr ha registrato quattro transazioni in uscita il 31 dicembre, per un totale di oltre 200 BTC dal valore, al tempo delle scrittura, di circa 3,6 milioni di dollari [19].
In un post successivo Dashjr ha pubblicato le transazioni sospette chiedendo aiuto alle autorità per risalre al colpevole, figura 5.2.


FIGURA 5.2: Transazioni denunciate

In prima analisi le transazioni truffaldine:

432ded946431a9612f09d73bd15ded045d11d1095ffdfe8d68306ea9b2e78930

c38a3210fbb758cfc41d9a64b7534b83aecca96f051231f15545e8e5c7365190

4b3cde50e2bce3d02e15b61957d2452e29f53d9a99e1ab14e83b6ec0f87fd851

50df1eab0bf2bd01999cea4fc531a65c17e1a285823c9ae4eab0feb7e21a11b6

contengono un elevato numero di UTXO in ingresso e la produzione di un singolo UTXO in uscita. Gli UTXO in ingresso sono assimilabili alle criptovalute possedute e rubate a Luke Dashjr mentre gli UTXO in uscita, trasferiti ad un singolo destinatario 1YAR6opJCfDjBNdn5bV8b5Mcu84tv92fa, sono le traccie di un possibile indirizzo appartenente ai criminali.

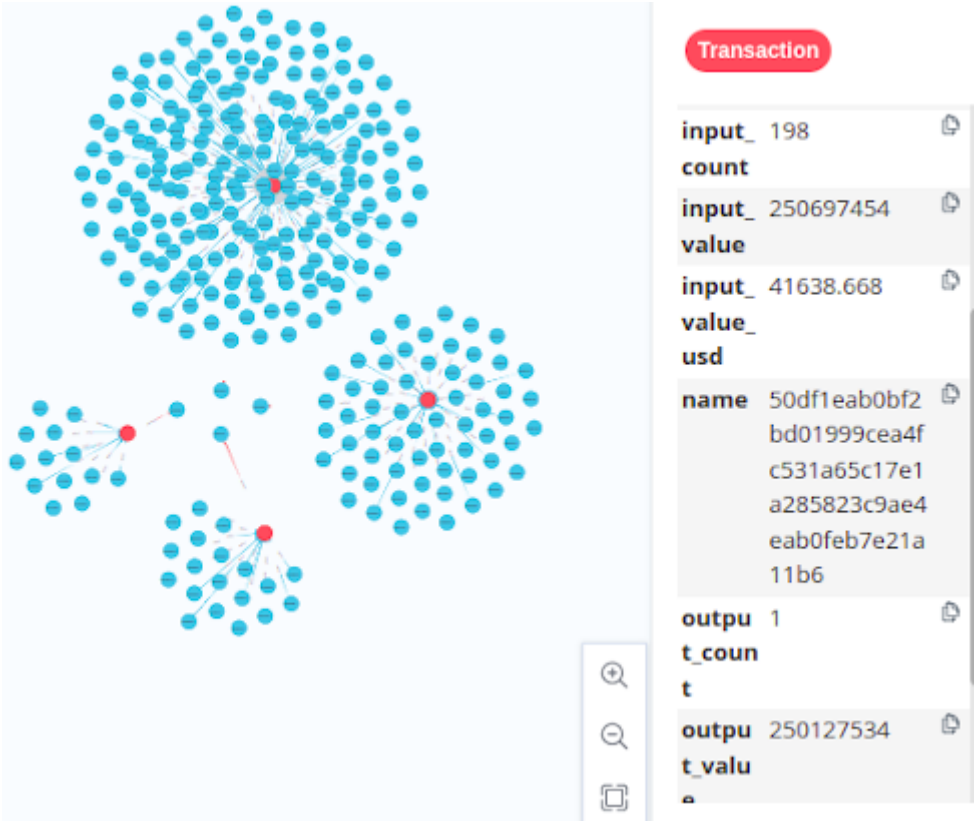


FIGURA 5.3: Transazioni iniziali caricate su GlockChain

Nella figura 5.3 sono presenti le transazioni denunciate, dove la parte centrale evidenzia gli output prodotti. I nodi UTXO di output sono stati esplorati in profondità, iterando sulla proprietà *spending_transaction_hash*, la quale contiene l'informazione sull'hash alla transazione successiva, ovvero la transazione dove sono stati spesi i suddetti UTXO.

Le iterazioni effettuate mirano a seguire il flusso di denaro e ad evidenziare pattern ricorrenti.
Durante il processo di espansione la priorità è stata data ad UTXO con un valore in satoshi elevato, ovvero con la maggior parte del denaro, e successivamente sono state esplorate le transazioni che includono UTXO con un valore in satoshi minore, figura 5.4.

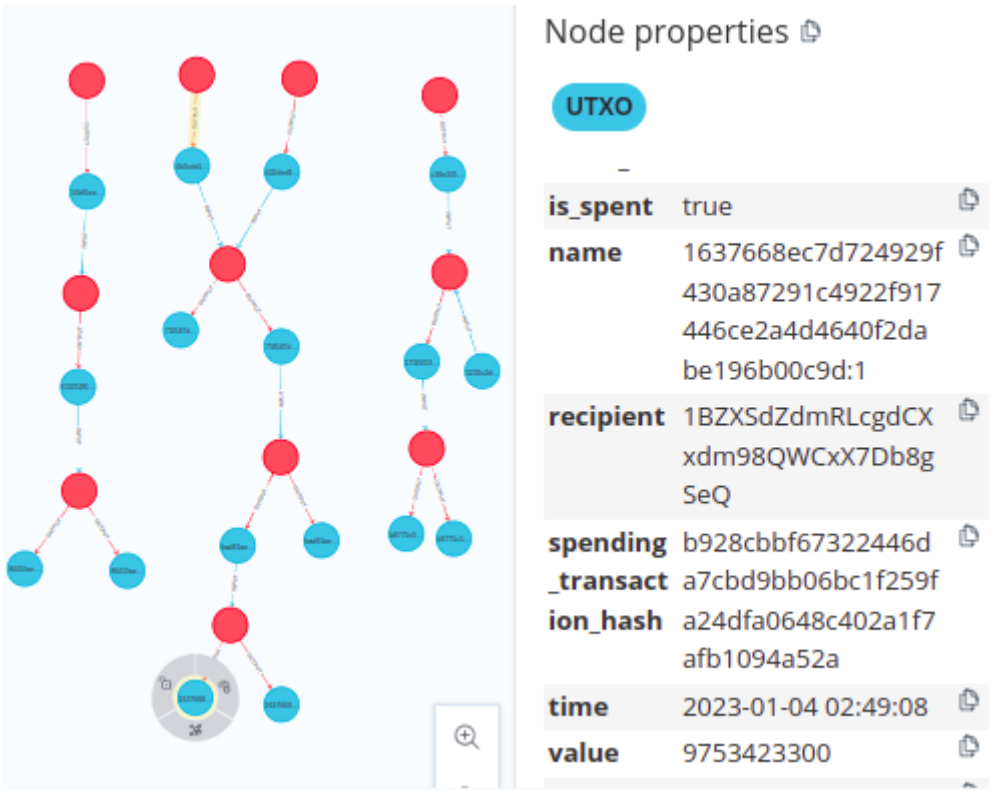


FIGURA 5.4: Espansione del grafo

Il numero delle transazioni successive a quelle sospette è notevole, poichè il flusso che determina lo spostamento dei satoshi evidenzia una catena di eventi che si ramifica in profondità.
Seguire il flusso di transazioni con il maggior numero di satoshi produce il grafo mostrato in figura 5.5 .
Grazie allo strumento Neo4j Bloom, che consente una visualizzazione chiara e gerarchica del grafo, si nota come le transazioni iniziali, in verde, vengono susseguite da transazioni caratterizzate da un singolo UTXO in input e due UTXO in output.

L'UTXO di input viene suddiviso in due nuovi UTXO: uno contenente la maggior parte dei Satoshi e l'altro contenente una quantità minore, figura 5.6. Questo meccanismo permette ai criminali di sottrarre il denaro, dall'ammontare totale, in maniera graduale, aumentando il numero delle transazioni e sottraendo una piccola quantità di satoshi in ogni iterazione. Tale evoluzione è una caratteristica ricorrente nel grafo e fà sorgere l'ipotesi di smurfing dei Bitcoin [20].

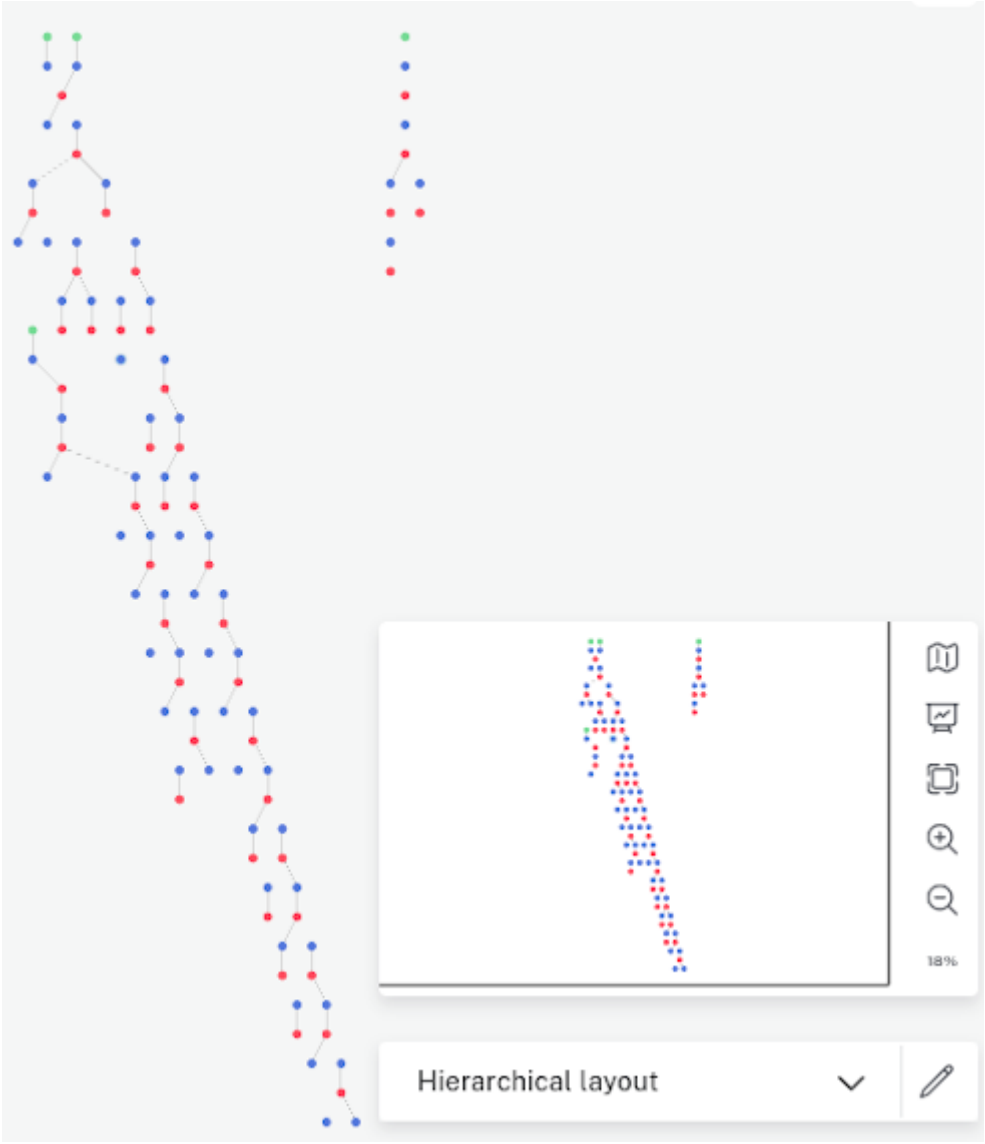


FIGURA 5.5: Catena parziale delle transazioni

## 5.2 Schemi Ricorrenti

Lo smurfing è una tecnica utilizzata per nascondere grandi somme di denaro in criptovalute suddividendo il loro valore in transazioni numerose, di piccolo importo e difficili da tracciare. Il processo inizia con un individuo che possiede una quantità significativa di criptovaluta che desidera utilizzare o spostare senza attirare l'attenzione. Invece di effettuare un'unica transazione di grande dimenzione, che potrebbe potenzialmente innescare controlli normativi o sollevare sospetti, la transazione originale viene divisa in molteplici transazioni più piccole, ciascuna delle quali al di sotto di una determinata soglia di segnalazione o progettata per eludere il rilevamento.
Queste transazioni di dimensioni ridotte vengono poi distribuite su più account, portafogli o indirizzi che possono appartenere alla stessa persona, oppure possono essere controllati da persone o entità differenti, come associati o complici.

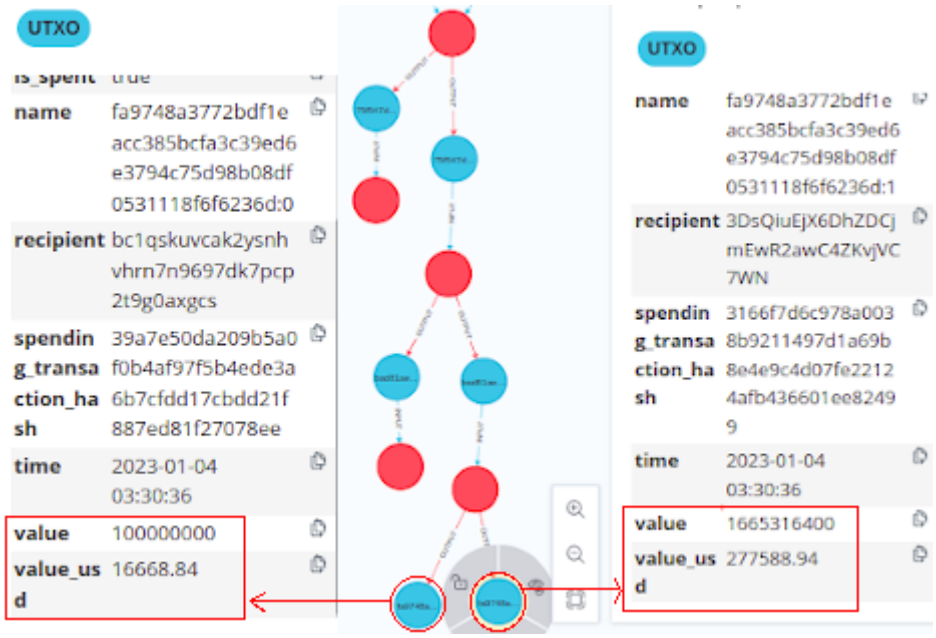


FIGURA 5.6: Esempio di divisione della somma di satoshi

Successivamente vengono implementate ulteriori tecniche per offuscare maggiormente la traccia delle transazioni come l'utilizzo di Mixers o di Layered Transactions. In questo modo dopo aver suddiviso e ridistribuito con successo i fondi, lo smurfer può ritirare i piccoli importi in modo da farli apparire legittimi.

Il grafo delle transazioni, in figura 5.7, mostra l'avanzamento dell'investigazione e la scoperta di un nuovo schema ricorrente. I criminali, dopo aver sottratto dal totale una quantità moderata di satoshi, hanno offuscato tale quantità con delle transazioni caratterizzate da pochi UTXO di input e una media di 30 o più UTXO di output.

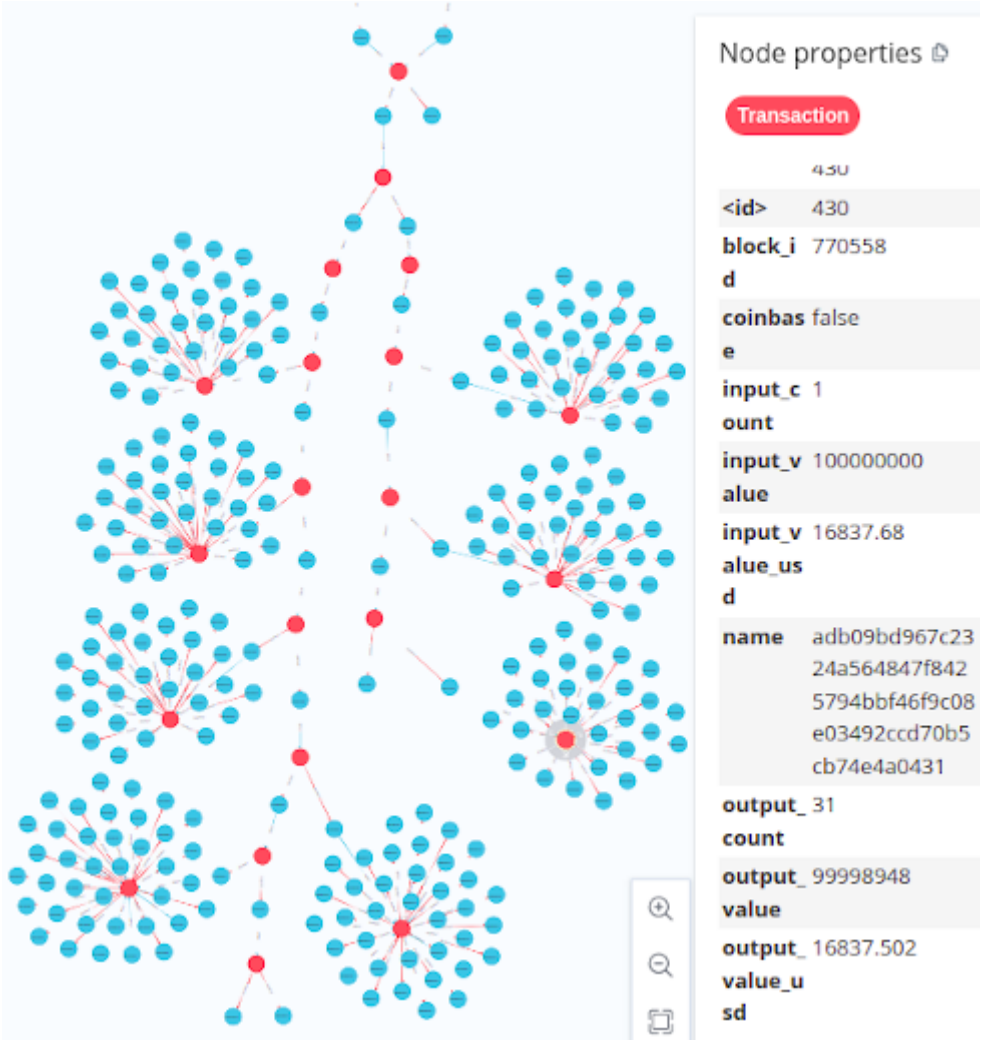


FIGURA 5.7: Smurfing delle transazioni

Riassumendo, dopo aver rubato il denaro i criminali hanno creato una catena di eventi caratterizzata da transazioni con un input e due output, figura 5.5. Nei due output prodotti i satoshi dell'input vengono divisi creando un UTXO contenete la maggior parte di essi ed un UTXO contenete una piccola parte, figura 5.6. L'UTXO con il valore in satoshi maggiore continua la catena creando

due nuovi output, metre l'UTXO con il valore in satoshi minore viene indirizzato in una transazione caratterizzata dalla produzione di 30 o più output, figura 5.7. Questi schemi si ripetono per tutto il grafo e terminano solamente quando la somma totale dei satoshi è stata completamente suddivisa in piccole parti e spesa in transazioni con un elevato numero di UTXO di output. Così facendo i criminali hanno diluito con efficacia i Bitcoin rendendo complesso l'ulteriore tracciamento dei fondi.

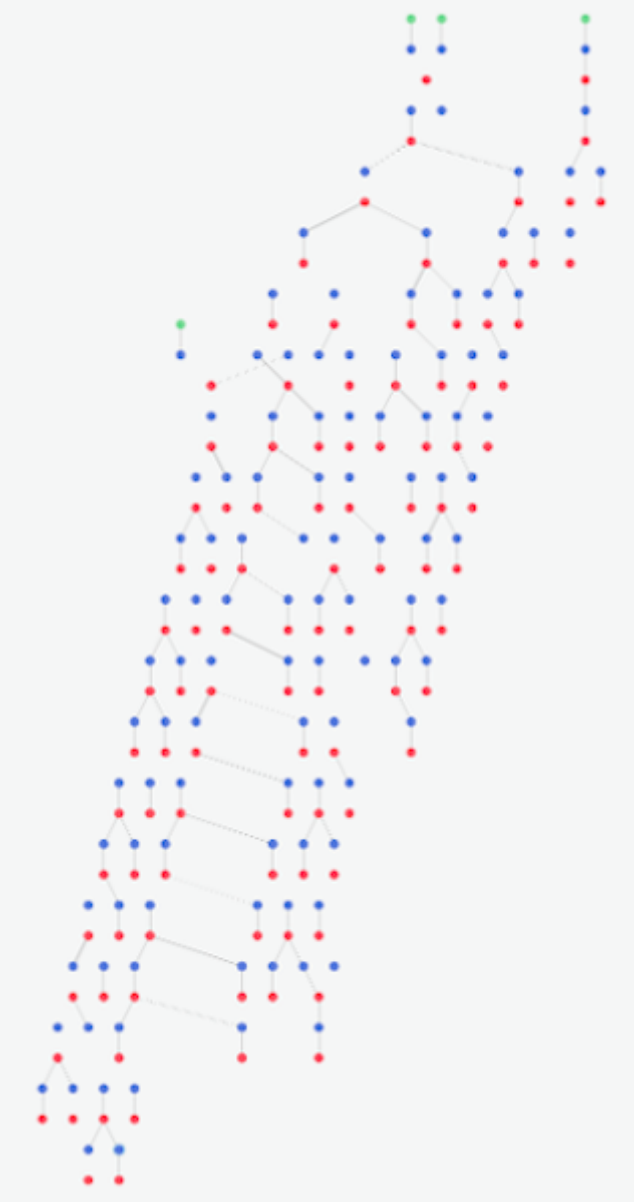

FIGURA 5.8: Vista complessiva delle grafo

## 5.3 Sviluppi futuri

Grazie al software è stato possibile tracciare il flusso delle transazioni successive al furto, mostrando visivamente come il valore in Satoshi sia stato distribuito per la rete Bitcoin. Inoltre, da una semplice analisi visiva, sono stati individuati due schemi ricorrenti, sfruttati per diffondere ed offuscare la provenienza e la destinazione del denaro.
Le potenzialità del software, dunque, sono promettenti e i possibili sviluppi futuri sono innumerevoli. Innanzittutto, le capacità implementate possono essere estese a tutte le criptovalute in modo da non limitare l'utilizzo del software ad una singola blockchian. La fase di esplorazione ed espansione del grafo potrebbe essere affiancata da script o funzionallità che agevolino e potenzino l'investigatore nell'analisi e nella scelta delle transazioni da esplorare.
Algoritmi di clustering, come il K-Means Clustering, vengono spesso utilizzati nella Transaction Graph Analysis per raggruppare transazioni, indirizzi o wallet correlati. L'integrazione del software con questi algoritmi, potrebbe aumentare le possibilità di trovare dei modelli o degli schemi.
Tecniche di machine learning potrebbero fornire al software la capacità di identificare pattern di comportamento in autonomia e di prevedere i futuri movimenti delle criptovalute.
Infine, l'applicazione dell'analisi off-chain alle informazioni estratte e messe in luce dal grafo potrebbe risolversi nella riduzione della pseudoanonimicità, favorendo il riconoscimento degli individui che effettuano le transazioni.
L'immaginazione è l'unico limite che ostacola la nostra capacità di trasformare le idee in soluzioni concrete ed è una qualità cardine nello sviluppo di software che consenta di esplorare nuove possibilità e di trovare soluzioni innovative ai problemi del mondo.

# Conclusioni

In questa tesi, è stato esplorato l'universo Bitcoin dalla nascita alla sua evoluzione. Le tecnologie implementate consentono di creare un ambiente sicuro, decentralizzato e trasparente per la gestione di transazioni finanziarie. Tuttavia, le sue proprietà positive vengono spesso sfruttate da malviventi per compiere ogni tipo di reato. L'analisi on-chain è uno strumento prezioso che permette di estrarre informazioni vitali per riconoscere questi criminali ed assicurarli alla giustizia. Tuttavia, è importante riconoscere che la blockchain analysis presenta molte sfide. La natura pseudonima della tecnologia rende difficile l'identificazione degli utenti. Inoltre, l'accesso ai dati on-chain può risultare costoso e richiedere competenze tecniche specifiche.
Nonostante le sfide è un campo in rapida crescita con il potenziale di rivoluzionare il modo in cui interagiamo con la tecnologia.
Grazie al suo continuo sviluppo e alle nuove tecniche messe in campo, l'analisi on chain ha il potenziale di rendere sempre più accessibile e sicura la convivenza umana con le blockchain assicurando, per una gamma sempre più ampia e diversificata di utenti, la fruizione di servizi innovativi ed affidabili .

# Bibliografia

[1] E. Hughes, «A Cypherpunk's Manifesto,» 1993. indirizzo: https://cdn.nakamotoinstitute.org/docs/cypherpunk-manifesto.txt.

[2] T. A. Mahler, «A Brief Prehistory Of Blockchain,» 2018. indirizzo: https://medium.com/thedarkside/a-brief-prehistory-of-blockchain-957bcd45604c.

[3] S. Nakamoto, «Bitcoin: A Peer-to-Peer Electronic Cash System,» 2008. indirizzo: https://bitcoin.org/bitcoin.pdf.

[4] W. Stallings, *Cryptography and Network Security - Principles and Practice*, 7th. Pearson, 2017, cap. 11 : Cryptographic Hash Funcions, pp. 348–350.

[5] W. Stallings, *Cryptography and Network Security - Principles and Practice*, 7th. Pearson, 2017, cap. 13 : Digital Signatures.

[6] A. M. Antonopoulos. «Mastering Bitcoin: Programming the Open Blockchain.» (2017), indirizzo: https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch04_keys.adoc.

[7] *Peer-to-peer*. indirizzo: https://en.wikipedia.org/wiki/Peer-to-peer.

[8] A. M. Antonopoulos. «Mastering Bitcoin: Programming the Open Blockchain.» (2017), indirizzo: https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch10_network.adoc.

[9] N. Furneaux, *Investigating Cryptocurrencies: Understanding, Extracting, and Analyzing Blockchain Evidence*. Wiley, 2018, cap. 3 : Understanding the Blockchian, pp. 40–51.

[10] N. Furneaux, *Investigating Cryptocurrencies: Understanding, Extracting, and Analyzing Blockchain Evidence*. Wiley, 2018, cap. 4 : Transactions, pp. 67–79.

[11] A. M. Antonopoulos, *Mastering Bitcoin: Programming the Open Blockchain*, 2ª ed. O'Reilly Media, Inc., 2017. indirizzo: https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch12_mining.adoc.

[12] *Choose Your Wallet*. indirizzo: https://bitcoin.org/en/choose-your-wallet.

[13] *BIP39: Seed phrase*. indirizzo: https://en.bitcoin.it/wiki/Seed_phrase.

[14] *BIP 32: Deterministic Wallets*. indirizzo: https://en.bitcoin.it/wiki/BIP_0032.

[15] Garante per la protezione dei dati personali, *Ransomware*, 2023. indirizzo: https://www.garanteprivacy.it/temi/cybersecurity/ransomware.

[16] Doubloin, «What is Bitcoin Transaction Graph?,» 2023. indirizzo: https://www.doubloin.com/learn/what-is-bitcoin-transaction-graph.

[17] Neo4j, *Neo4j Getting Started Guide*. indirizzo: https://neo4j.com/docs/getting-started/.

[18] L. Fontana, *GlockChain*, 2024. indirizzo: https://github.com/Luigi-Fontana-96/GlockChain.

[19] «Bitcoin Developer PGP Exploit.» (2023), indirizzo: https://www.theblock.co/post/198688/bitcoin-developer-pgp-exploit.

[20] ImmuneBytes, «Use of Smurfing in Crypto Laundering,» *ImmuneBytes Blog*, 2023. indirizzo: https://www.immunebytes.com/blog/use-of-smurfing-in-crypto-laundering/.