Blockchain CheatSheet - Uso Tecnico

Tempo di Lettura: 5 m

Indice

§ Indirizzi

Casi d'Uso

- Firmare una transazione con una chiave pubblica per identificare e convalidare i dati.
- Chiunque possieda la chiave pubblica può identificare e convalidare i dati.

I Passaggi

1. **Generazione delle Coppie di Chiavi**:
   - Creare una ***Chiave Privata***: `256 bit o 64 caratteri esadecimali`
     - Generazione casuale.
   - Derivare la ***Chiave Pubblica Base***: `512 bit o 128 caratteri esadecimali`
     - Utilizzare la ***Chiave Privata*** con l' *Elliptic Curve Digital Signature Algorithm* (Algoritmo => x*coordinate-256bit + y_coordinate-256bit = 512 bit* **Chiave Pubblica Base**_).
2. **Hashing (Ethereum)**:
   - Hash della ***Chiave Pubblica***: `Da 512 bit a 256 bit o 64 caratteri esadecimali`
     - Hash della ***Chiave Pubblica Base*** con *Kekkak-256* o *Sha-3*.
3. **Generazione dell'Indirizzo Pubblico (Ethereum)**:
   - Creare un ***Indirizzo Pubblico***: `Da 64 caratteri esadecimali a 42 caratteri esadecimali`
     - Prendere gli ultimi 40 caratteri esadecimali (20 byte) e prefissare con 0x per ottenere 42 caratteri esadecimali.

---

§ Cryptotransazioni

Analogia

Supponiamo che le parti **A**, **B** e **C** abbiano ciascuna una *cassetta di sicurezza* che contiene contenuti che viaggiano attraverso il Sistema di Protocollo Blockchain, il quale applica le regole di funzionamento. Queste *cassette di sicurezza* hanno una fessura che accetta solo contenuti in entrata, e l'unico modo per recuperare il contenuto è con la chiave privata del proprietario.

Meccanica delle Transazioni

A Invia -> a B Dati o Criptovaluta

1. **B** Crea l'***Indirizzo Pubblico*** e la ***Chiave Pubblica*** dalla ***Chiave Privata***:
   - ***Chiavi Private*** di **B** :: ***Indirizzo Pubblico*** e ***Chiave Pubblica*** di **B**.
2. **B** Invia l'***Indirizzo Pubblico*** -> a **A** (*L'indirizzo pubblico può cambiare per ogni transazione*).
   - ***Indirizzo Pubblico*** di **B** -> a **A**.
3. **A** aggiungerà l'***Indirizzo Pubblico*** di **B** e i dati o l'importo a un messaggio di "Transazione":
   - **A** Inizializza la Transazione :: ***Indirizzo Pubblico*** di **B** e Contenuto.
4. **A** firmerà la transazione con la ***Firma Digitale***:
   - ***Firma Digitale*** :: Derivata dalla Chiave Privata di **A** con l' *Elliptic Curve Digital Signature Algorithm* (x_coordinate-256bit + y_coordinate-256bit).
5. La Transazione di **A** è *Proposta* dal protocollo blockchain nel *Memory Pool*:
   - ***Validazione*** :: I miner tentano di validare la transazione includendola in un blocco del memory pool.

Validazione della Proposta

B poi Invia -> a C

- È necessario verificare prima di imprimere la transazione nella Blockchain che **B** abbia effettivamente il contenuto necessario da inviare nuovamente: **Transazione di B** è Inviata -> al **Memory Pool** della Blockchain e poi il Protocollo Invia -> a **C**.

Cryptotransazioni Bitcoin in Dettaglio

Bitcoin vs Ethereum

- **Bitcoin**: Ogni transazione deve essere considerata come un contenitore di criptovaluta unica non mischiata con le altre.
- **Ethereum**: A differenza di Bitcoin, utilizza un sistema contabile che tiene traccia del saldo totale.

Gestione delle Transazioni

Le criptovalute, essendo legate ai contenitori di transazione che chiameremo *X*-Trsct-C*n* (*X* = ID, Trsct = Transazione, C*n* = Numero del Contenitore), devono essere gestite manipolando il contenitore.

A Invia 10 Bitcoin -> a B da un Contenitore di Transazione che contiene 20 Bitcoin

1. **B** Crea l'***Indirizzo Pubblico*** e la ***Chiave Pubblica*** dalla ***Chiave Privata***.
2. **B** Invia l'***Indirizzo Pubblico*** -> a **A** (*L'indirizzo pubblico può cambiare per ogni transazione*).
3. **A** aggiungerà l'***Indirizzo Pubblico*** di **B** e l'importo a un messaggio di "Transazione".
4. Il *Nuovo Contenitore di Transazione Vuoto* (**A**-Trsct-C4) prenderà un input e invierà uno o due output, l'importo e il cambio eventuale:
   - L'input si basa sui contenitori di transazione che hanno la criptovaluta non spesa o UTXO (Unspent Transaction Output) che copre l'importo della Nuova Transazione.
     - A-Trsct-C1 = 10 Bitcoin
     - A-Trsct-C2 = 30 Bitcoin -> Input
     - A-Trsct-C3 = 5 Bitcoin
   - Il primo output sarà l'importo della Nuova Transazione.
     - A-Trsct-C4 = 20 Bitcoin -> Output a B-Trsct-C1
   - L'output opzionale sarà il cambio che viene restituito al mittente A.
     - A-Trsct-C4 = 10 Bitcoin -> Output a A-Trsct-C4
     - !!!
     - A-Trsct-C2 = 30 Bitcoin viene quindi Distrutto
5. **A** firmerà la transazione con la ***Firma Digitale***.
6. La Transazione di **A** è *Proposta* dal protocollo blockchain nel *Memory Pool*.
7. Validazione della *Proposta*.

Validazione del Proof of Work dei Miner

Diagramma

css

Copia codice

```
`Transazioni/dati -> Gruppo di Transazioni di 80 byte
v                                                    Intestazione del
Blocco                                                          |
v                                                 * Ricerca Nonce
|                                                               v
Calcolo dell'algoritmo di hash
|                                          ----------------------------
-----                                      |
|                              Hash Valido                    Hash Non-
Valido                                     |
|                 # Blocco validato e aggiunto          * Incremento Nonce
```

```
alla Blockchain                    Ripetizione del                    &
Ricompensa al Minatore              Processo`
```

Struttura del Blocco Blockchain

Nel contesto della blockchain, i miner creano blocchi con una struttura specifica. Un'intestazione di blocco tipica di Bitcoin è di 80 byte e include i seguenti elementi:

- 4 byte: numero di versione
- 32 byte: hash del blocco precedente
- 32 byte: radice di Merkle (hash delle transazioni nel blocco)
- 4 byte: timestamp
- 4 byte: obiettivo di difficoltà
- 4 byte: nonce

> Di solito, le uniche differenze tra i tentativi di hashing dei miner sono:
>
> - L'hash dei dati (la prima parte del quale è la ricompensa per il miner).
> - Il timestamp (che può variare non solo per posizione, ma anche per il numero di tentativi di trovare il nonce).
> - Il nonce stesso.
> - Inoltre, l'ordine in cui i dati sono raggruppati può variare tra i miner.

---

§ Scalabilità

Livelli

- **Layer 0**: Internet così come lo conosciamo.
- **Layer 1**: Le transazioni del Layer 1 della blockchain sono più lente rispetto ai metodi tradizionali, fino a 10 minuti per un regolamento.
- **Layer 2**: Wallet per transazioni piccole più veloci.

Layer 2 Lightning Network

Descrizione: Il Lightning Network è una soluzione off-chain che funziona come un canale di pagamento, costruito su una struttura di rete che collega gli utenti. Permette di elaborare le transazioni senza registrare ogni transazione sulla blockchain di Bitcoin.

Cosa Risolve: Aumenta significativamente la velocità delle transazioni utilizzando un sistema di doppia firma come accordo di scambio.

Come Funziona: Ogni volta che i clienti devono effettuare un pagamento, entrambe le parti inviano una transazione come descritto nella Gestione delle Transazioni. Il mittente imposta l'importo dovuto e il ricevente imposta una transazione con un valore quasi nullo. La richiesta di pagamento viaggia attraverso la rete, cercando il percorso più breve di canali connessi per raggiungere il destinatario. Ogni canale detiene un saldo che può essere trasferito tra le parti coinvolte, e solo l'apertura e la chiusura dei canali sono registrati sulla blockchain.

Suggerimento per il Proseguimento

Blockchain CheatSheet - Consensus

Autore: Kenneth Boldrini