Generazione degli Indirizzi P2PKH

Davide3011

January 2025

1 Introduzione

Gli indirizzi P2PKH (Pay-to-PubKey-Hash) rappresentano uno dei formati di indirizzo più diffusi e fondamentali nella rete Bitcoin. Introdotti nel 2009, poco dopo il lancio del protocollo Bitcoin, gli indirizzi P2PKH sono stati progettati come un miglioramento rispetto al formato iniziale **Pay-to-Public-Key (P2PK)**. Nei primi blocchi della blockchain (es. blocco 0, noto come Genesis Block), le transazioni utilizzavano indirizzi P2PK, che inviavano fondi direttamente alla chiave pubblica del destinatario. Questo approccio, sebbene funzionale, presentava una vulnerabilità significativa: l'esposizione della chiave pubblica poteva rappresentare un rischio per la sicurezza nel lungo termine.

Con l'adozione di P2PKH, che iniziò a essere ampiamente utilizzato nei primi blocchi successivi al Genesis Block, Bitcoin introdusse un meccanismo più sicuro e compatto. Gli indirizzi P2PKH utilizzano l'**hash della chiave pubblica** (tramite SHA-256 e RIPEMD-160) per nascondere la chiave pubblica sottostante, riducendo il rischio di attacchi crittografici futuri.

Gli indirizzi P2PKH sono identificabili dal loro prefisso `1` nella rete principale (mainnet) e dai prefissi `m` o `n` nella rete di test (testnet). Essi sono generati seguendo un processo ben definito, che include l'applicazione di funzioni crittografiche e la codifica del risultato finale nel formato **Base58Check** per migliorare la leggibilità.

In questo documento, analizzeremo il processo di generazione degli indirizzi P2PKH in dettaglio, considerando due approcci principali:

- Utilizzando una **chiave pubblica non compressa**, lunga 65 byte, che contiene informazioni dettagliate sul punto della curva ellittica generato dalla chiave privata.
- Utilizzando una **chiave pubblica compressa**, lunga 33 byte, che rappresenta un formato più compatto ma equivalente in termini di funzionalità.

Entrambi gli approcci condividono la stessa struttura generale del formato dell'indirizzo, che include:

- L'applicazione di **SHA-256** e **RIPEMD-160** alla chiave pubblica per ottenere l'hash della chiave pubblica.
- L'aggiunta di un **prefisso di rete** per identificare la rete utilizzata (es. `0x00` per mainnet).

- La generazione di un **checksum**, utilizzando doppio SHA-256, per rilevare errori di digitazione o alterazioni accidentali.
- La codifica del risultato finale nel formato **Base58Check**, che produce un indirizzo leggibile dall'utente.

Grazie a questa struttura, gli indirizzi P2PKH hanno maggiore sicurezza rispetto ai loro predecessori P2PK, diventando lo standard per le transazioni nella rete Bitcoin. Sebbene siano ancora ampiamente utilizzati, gli indirizzi P2PKH risultano meno efficienti rispetto ai formati più moderni in termini di dimensioni delle transazioni e costi delle commissioni.

2 Chiave Privata e Pubblica

2.1 Generazione della Chiave Privata

La chiave privata è un numero casuale di 256 bit generato utilizzando un generatore di numeri casuali crittograficamente sicuro (CSPRNG). Ad esempio:

```
Chiave privata:
e9873d79c6d87dc0fb6a5778633389e486f4dd9c0677d39b8a2e0f140d4a2d07
```

3 Formato WIF (Wallet Import Format)

Il formato WIF (*Wallet Import Format*) è una rappresentazione leggibile della chiave privata. Questo formato è comunemente utilizzato per importare o esportare chiavi private in wallet Bitcoin. Il WIF è essenzialmente una codifica Base58Check della chiave privata, con alcuni passaggi aggiuntivi per garantire la sicurezza e l'identificabilità della rete.

3.1 Passaggi per Calcolare il Formato WIF

Il processo per ottenere il WIF è il seguente:

1. **Aggiunta del prefisso della rete:**
   - Per **mainnet**, il prefisso è `0x80`.
   - Per **testnet**, il prefisso è `0xEF`.

   Al valore della chiave privata (32 byte) viene aggiunto il prefisso della rete.
2. **Aggiunta del byte di compressione (opzionale):**
   - Se la chiave pubblica associata è **compressa**, viene aggiunto il byte `0x01` alla fine.
   - In caso contrario, non viene aggiunto alcun byte.
3. **Calcolo del checksum:**

- Si applica SHA-256 due volte al risultato dei passaggi precedenti.
- I primi 4 byte del risultato sono il checksum.

4. **Codifica in Base58Check:**
   - Il risultato finale è ottenuto concatenando:
     - Prefisso della rete + Chiave privata + Byte di compressione (se applicabile) + Checksum.
   - Questo valore viene codificato nel formato Base58Check per generare il WIF leggibile.

3.2 Esempio di Calcolo del WIF

- **Chiave privata (esadecimale):**

  `e9873d79c6d87dc0fb6a5778633389e486f4dd9c0677d39b8a2e0f140d4a2d07`

- **Chiave privata con prefisso (mainnet):**

  `80e9873d79c6d87dc0fb6a5778633389...0f140d4a2d07`

- **SHA-256 (prima applicazione):**

  `5b3e9a65fbdc014f94ecf88aa45dfd2f2f577d648d3bbf9b92321cb16f7a70e4`

- **SHA-256 (seconda applicazione):**

  `6028f25e5a68b6f731f40fc5c884b58c8d4ef664aab1db24936e11d7d7267a2b`

- **Checksum:**

  `6028f25e`

- **Base58Check finale:**

  `5HueCGU8rMjxEXxiPuD5BDu...`

Con il formato WIF, l'importazione e l'esportazione di chiavi private diventa più semplice e leggibile, mantenendo la sicurezza e la compatibilità.

3.3 Derivazione della Chiave Pubblica

La chiave pubblica viene derivata dalla chiave privata utilizzando l'algoritmo della curva ellittica `secp256k1`. Ecco il processo:

Curva Ellittica secp256k1

La curva è definita dall'equazione:

$$y^2 = x^3 + 7 \mod p$$

dove:

- $p = 2^{256} - 2^{32} - 977$: modulo primo.
- $G = (x_G, y_G)$: punto generatore della curva.
- n: ordine della curva (numero totale di punti).

Moltiplicazione Scalare

La chiave pubblica P è calcolata come:

$$P = d \cdot G$$

dove:

- d: chiave privata.
- G: punto generatore.
- $P = (x_P, y_P)$: punto risultante sulla curva.

Operazioni Coinvolte:

1. **Somma di Punti:** Dati due punti $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$, la somma $P_3 = P_1 + P_2$ si calcola come:
$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \mod p$$
$$x_3 = \lambda^2 - x_1 - x_2 \mod p, \quad y_3 = \lambda(x_1 - x_3) - y_1 \mod p$$
2. **Raddoppio di un Punto:** Se $P_1 = P_2$, si usa:
$$\lambda = \frac{3x_1^2}{2y_1} \mod p$$
$$x_3 = \lambda^2 - 2x_1 \mod p, \quad y_3 = \lambda(x_1 - x_3) - y_1 \mod p$$
3. **Moltiplicazione Scalare:** Si calcola $d \cdot G$ utilizzando l'algoritmo **double-and-add**:
   - Espandi d in binario.
   - Per ogni bit di d: raddoppia il punto corrente e, se il bit è 1, aggiungi G.

Formato della Chiave Pubblica

- **Non compressa (65 byte):** Inizia con il prefisso `0x04`, seguito da 32 byte per x e 32 byte per y.
- **Compressa (33 byte):** Inizia con `0x02` o `0x03`, a seconda che y sia pari o dispari, seguito da 32 byte per x.

4 Generazione dell'Indirizzo P2PKH

4.1 Con Chiave Pubblica Non Compressa

Il processo per generare un indirizzo P2PKH con una chiave pubblica non compressa include i seguenti passaggi. Viene fatto riferimento alle differenze tra mainnet e testnet:

1. **Hash della Chiave Pubblica:**
   - La chiave pubblica non compressa (65 byte, prefisso `0x04`, seguito da 32 byte per x e 32 byte per y) viene sottoposta a due funzioni hash successive:
     - **SHA-256:** Calcola un hash di 256 bit della chiave pubblica.
     - **RIPEMD-160:** Calcola un hash di 160 bit sul risultato del SHA-256, ottenendo il *Public Key Hash (PKH)*.
   - Questo passaggio è identico per entrambe le reti.
2. **Aggiunta del Prefisso di Rete:**
   - Al Public Key Hash viene aggiunto un prefisso per identificare la rete:
     - `0x00` per **mainnet**.
     - `0x6F` per **testnet**.
3. **Calcolo del Checksum:**
   - Viene calcolato un checksum per rilevare eventuali errori di trascrizione. Si applicano due volte la funzione **SHA-256** al risultato del passaggio precedente (Prefisso + PKH).
   - I primi 4 byte del risultato finale vengono utilizzati come checksum.
   - Questo passaggio è identico sia per mainnet che per testnet.
4. **Codifica Base58Check:**
   - Il risultato (Prefisso + PKH + Checksum) viene codificato in **Base58Check** per produrre l'indirizzo leggibile dall'utente.
     - Gli indirizzi su **mainnet** iniziano con `1`.
     - Gli indirizzi su **testnet** iniziano con `m` o `n`.

4.2 Con Chiave Pubblica Compressa

Il processo per generare un indirizzo P2PKH con una chiave pubblica compressa segue gli stessi passaggi descritti sopra. La differenza principale riguarda il formato della chiave pubblica:

- **Chiave pubblica compressa (33 byte):**
  - Inizia con `0x02` o `0x03`, a seconda che la coordinata y sia pari o dispari, seguito da 32 byte per la coordinata x.
  - Questo formato riduce la dimensione della chiave pubblica rispetto alla versione non compressa (65 byte).
- **Hashing e Prefissi:**
  - I passaggi di hashing, aggiunta del prefisso, calcolo del checksum e codifica Base58Check sono identici per entrambe le reti (mainnet e testnet).
- **Risultato Finale:**
  - Gli indirizzi generati su **mainnet** iniziano con `1`.
  - Gli indirizzi generati su **testnet** iniziano con `m` o `n`.

5 Esempi

5.1 Esempio con Chiave Pubblica Non Compressa

- **Chiave Pubblica Non Compressa:**

  `04bfcab1cd4c9b5b2c1a62d27f5a6f9b6e9b1c8c1e4f85...`

- **Hash SHA-256:**

  `f54d3f1c2b3e45f1d2d50f4c91b831...`

- **Hash RIPEMD-160:**

  `76a91488ac...`

- **Indirizzo Finale (Mainnet):**

  `1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa`

5.2 Esempio con Chiave Pubblica Compressa

- **Chiave Pubblica Compressa:**

  `02c72e0f5c3b9d6f9301c5df30e0dbe234b61f...`

- **Hash SHA-256:**

  `a99f3f1b2d4e57f1d5d90f3c91b819...`

- **Hash RIPEMD-160:**

  `76a914abcd...`

- **Indirizzo Finale (Mainnet):**

  `1BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2`