Generazione di una Chiave Privata e Pubblica per Bitcoin (P2PK)

Davide3011

January 2025

Introduzione

Bitcoin utilizza un sistema di crittografia basato sulla matematica delle curve ellittiche per generare coppie di chiavi private e pubbliche. Questo sistema garantisce sicurezza, integrità e verificabilità delle transazioni. Il formato `P2PK` (*Pay-to-PubKey*) rappresenta uno dei primi metodi utilizzati per bloccare i fondi, in cui la chiave pubblica viene direttamente inclusa nella transazione.

Di seguito viene descritto il processo dettagliato per generare una chiave privata, calcolare la chiave pubblica e creare uno script `P2PK`.

1. Generazione della Chiave Privata

La chiave privata è un numero casuale scelto nell'intervallo:

$$1 \leq d \leq n - 1$$

dove:

- n è l'ordine del gruppo della curva ellittica `secp256k1`, che è un valore fisso pari a:

$$n = \texttt{0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141}$$

- d rappresenta la chiave privata ed è generato in modo pseudocasuale, utilizzando generatori casuali crittograficamente sicuri.

La chiave privata è generalmente un numero di 256 bit (32 byte) rappresentato in formato esadecimale. Esempio:

$$d = \texttt{0x1E99423A4EDF2F72B5782C3C1F7A1A72B83E0AB24035E1DFFABFA55A87F0C88F}$$

Importanza della casualità. La sicurezza di Bitcoin si basa sulla difficoltà di indovinare una chiave privata. Un generatore casuale debole o prevedibile può compromettere la sicurezza dei fondi.

2. La Curva Ellittica secp256k1

La curva ellittica `secp256k1` è definita dall'equazione:

$$y^2 \equiv x^3 + 7 \pmod{p}$$

dove:

- $p = 2^{256} - 2^{32} - 977$ è un grande numero primo che definisce il campo finito su cui opera la curva.
- Le operazioni sulla curva seguono le regole della matematica modulare, garantendo che tutte le operazioni siano effettuate all'interno di un intervallo definito.

Il Punto Generatore G. La curva definisce un punto generatore G, un punto fisso e noto a tutti, che rappresenta il punto di partenza per generare chiavi pubbliche. Le coordinate di G sono:

$$G = (x_G, y_G)$$

con:

$$x_G = \texttt{0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798}$$

$$y_G = \texttt{0x483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8}$$

3. Calcolo della Chiave Pubblica

La chiave pubblica P viene calcolata come:

$$P = d \cdot G$$

dove:

- d è la chiave privata.
- G è il punto generatore della curva ellittica.
- $P = (x_P, y_P)$ rappresenta la chiave pubblica come coppia di coordinate.

Moltiplicazione scalare. L'operazione $d \cdot G$ è una somma ripetuta del punto G su se stesso d volte, utilizzando le regole della curva ellittica:

- L'addizione di due punti sulla curva segue una formula deterministica che combina le loro coordinate.
- La moltiplicazione scalare è un'operazione computazionalmente semplice, ma il processo inverso (determinare d conoscendo P) è praticamente impossibile grazie al problema del logaritmo discreto.

4. Compressione della Chiave Pubblica

La chiave pubblica può essere rappresentata in due formati:

- **Non compressa**: include entrambe le coordinate x_P e y_P, precedute dal prefisso `0x04`:

$$\texttt{0x04} \,||\, x_P \,||\, y_P$$

- **Compressa**: include solo x_P e un prefisso che indica il segno di y_P:

$$\texttt{0x02} \text{ se } y_P \text{ è pari, altrimenti } \texttt{0x03} \,||\, x_P$$

Scelta del formato. Il formato compresso riduce l'ingombro della chiave pubblica nelle transazioni, migliorando l'efficienza del protocollo.

5. Creazione dello Script P2PK

Uno script `P2PK` include direttamente la chiave pubblica nella transazione. Il formato è:

$$\langle \text{chiave\_pubblica} \rangle \, \texttt{OP\_CHECKSIG}$$

dove:

- $\langle$chiave_pubblica$\rangle$ è la rappresentazione binaria della chiave pubblica.
- `OP_CHECKSIG` verifica che la firma corrisponda alla chiave pubblica inclusa.

Questo script è oggi considerato obsoleto e poco efficiente, poiché non protegge contro attacchi di riproduzione (*replay attacks*).

6. Riassunto del Processo

1. Generare un numero casuale d come chiave privata.
2. Calcolare la chiave pubblica P moltiplicando d per G.
3. Rappresentare la chiave pubblica in formato compresso o non compresso.
4. Inserire la chiave pubblica nello script `P2PK`.

7. Considerazioni di Sicurezza

- **Protezione della chiave privata:** La chiave privata non deve mai essere condivisa o archiviata in modo insicuro.
- **Uso di generatori casuali sicuri:** Evitare generatori deboli che potrebbero generare chiavi private prevedibili.
- **Obsolescenza di P2PK:** Gli script `P2PKH` (*Pay-to-PubKey-Hash*) e `P2WPKH` (*Pay-to-Witness-PubKey-Hash*) sono oggi standard per migliorare la sicurezza.

Nota Finale

La generazione di chiavi private e pubbliche, insieme alla comprensione degli script Bitcoin, è fondamentale per sviluppare una conoscenza approfondita del protocollo. Sebbene gli script `P2PK` siano storicamente rilevanti, la loro applicazione pratica è limitata ai contesti di studio.