Principles of Cryptocurrency Design

Appunti ed Esercizi

Francesco Pasquale

4 maggio 2026

Nella lezione di oggi abbiamo implementato la classe `Addr` e l'abbiamo usata per generare indirizzi Bitcoin a partire da numeri a 256 bit. Gli esercizi di questa nota si riferiscono al codice scritto in aula, che potete scaricare qui: `https://www.mat.uniroma2.it/~pasquale/dida/aa2526/pcd/pcd260504.zip`

Esercizio 1. Usando la libreria `secrets` di python (o un altro generatore sicuro di numeri casuali) generare una chiave privata tale che l'indirizzo Bitcoin mainnet associato alla chiave inizi con `1PCD`.

Esercizio 2. Verificare che uno degli indirizzi contenuti negli output script P2PKH della transazione b35d91a71f226ba961162ca18f321b4d9aada8a0e722430ad3d2d2e4dda9a2c0,
l'indirizzo `1CLJKodMLHhAwiDYFWDiwmz31cMfhqKnEc`, è l'hash di una chiave pubblica non compressa che ha come come chiave privata lo `sha256` della stringa `Francesco`. Quella transazione ha 500 output del tipo P2PKH tutti con lo stesso ammontare. Scrivere un programma *brute-force* che cerchi di individuare se in quella transazione ci sono altri indirizzi che hanno come chiave privata l'hash `sha256` di altri nomi.

Esercizio 3. Ottenere dei bitcoin testnet tramite qualche faucet, provare a eseguire delle transazioni verso indirizzi con chiavi private facilmente individuabili tramite brute-force e vedere quanto tempo "resistono" prima che qualche bot ne individui la chiave privata e li spenda.

Esercizio 4. A lezione abbiamo scritto un metodo WIF per la classe ADDR, che restituisce la chiave privata in formato WIF, e l'abbiamo usato per importare la chiave segreta in un wallet.

Scrivere un metodo di classe PARSE_WIF che legga una stringa di testo in *base58*, verifichi che gli ultimi quattro byte sono un *checksum* corretto dei glia altri byte secondo il formato WIF e restituisca l'oggetto corrispondente della classe ADDR.