Principles of Cryptocurrency Design

Appunti ed Esercizi

Francesco Pasquale

10 aprile 2025

Esercizio 1. Il `target` attuale (blocco n. `891953`) della blockchain di Bitcoin espresso in decimale è $151813 \cdot 2^{160}$.

1. Sapendo che il `target` è impostato in modo che, in media, viene creato un nuovo blocco ogni dieci minuti, stimare l'*hash rate* h (ossia il numero di hash per secondo) che i miner stanno complessivamente eseguendo attualmente.

2. Supponete che un extraterrestre arrivi sulla terra con un *hardware* che riesce a fare k volte più hash al secondo di quanti ne fanno tutti i miner attuali messi insieme, per qualche $k > 1$, e decida di voler usare il suo hardware per riscrivere l'intera blockchain di Bitcoin a partire dal *genesis block*. Assumendo che l'hash rate degli altri miner rimanga costante, calcolare approssimativamente quanto tempo passerebbe prima che, in accordo al protocollo Bitcoin, tutti riconoscano la blockchain dell'extraterreste come quella valida.

Esercizio 2. Assumendo che la funzione crittografica `sha256` sia un *random oracle*, qual è la probabilità che il prossimo blocco impieghi più di k minuti per essere creato, per $k > 10$?

1 Transazioni

Ogni blocco della blockchain di Bitcoin è formato da `header` e `data`. Nella scorsa lezione abbiamo affrontato in dettaglio il contenuto dell'`header`, vediamo ora il contenuto del `data`. Ogni elemento in `data` è una *transazione*.

1.1 Input e output

Una *transazione* `tx`, è una struttura formata da uno o più *input* e uno o più *output* (anche da un numero di *versione* e da un *locktime*, ma per il momento non ce ne preoccupiamo).

Un `output` contiene due campi: un `ammontare`, ossia un numero intero che corrisponde all'ammontare di quell'output e un `locking_script`.

Un `input` è una struttura formata da (1) un puntatore all'output di una precedente transazione, indicato con la coppia (`prev_tx`, `prev_id`), dove `prev_tx` è la transazione e `prev_id` è l'indice dell'output all'interno della transazione e (2) un `unlocking_script` (c'è anche un *sequence number*, ma non ce ne occupiamo). Diciamo che una transazione $\texttt{tx}_a$ *spende* output $\texttt{tx}_b\texttt{.output}$ di una precedente transazione $\texttt{tx}_b$ se uno degli input di $\texttt{tx}_a$ punta all'output $\texttt{tx}_b\texttt{.output}$.

1.2 Script

Discuteremo in dettaglio gli script in una delle prossime lezioni. Per il momento diciamo solo che i due script, `locking_script` nell'output e `unlocking_script` nell'input, sono dei veri e propri "programmi" scritti in un linguaggio che tutti i nodi della rete possono interpretare. Data una transazione `tx` e un suo input `tx.input`, se la concatenazione dello *unlocking script* nell'input,

`tx.input.unlocking_script`, con il *locking script* contenuto nell'output della transazione precedente puntato da quell'input, `tx.input.(prev_tx, prev_id).locking_script` forma un programma che restituisce TRUE, allora l'input `tx.input` è *valido*. Una transazione `tx` è valida se (1) tutti i suoi input puntano a degli output che non sono già stati spesi da qualche altra transazione e sono validi, e (2) se la somma degli `ammontare` degli output, $\sum$ `tx.output.ammontare` è minore o uguale alla somma degli `ammontare` degli output puntati dagli input, $\sum$ `tx.input.(prev_tx, prev_id).ammontare`. La differenza, maggiore o uguale a zero, fra questi due valori è la *transaction fee* della transazione `tx`.

$$\texttt{tx.fee} = \sum_{\texttt{tx.input}} \texttt{tx.input.(prev\_tx, prev\_id).ammontare} - \sum_{\texttt{tx.output}} \texttt{tx.output.ammontare}$$

1.3 Transazione *coinbase*

Abbiamo detto che gli input di ogni transazione devono puntare ad output di transazioni precedenti. C'è una eccezione a questa regola (deve esserci, altrimenti andremmo a ritroso all'infinito...): la prima transazione di ogni blocco, cosiddetta *coinbase transaction*, non è soggetta a questo vincolo. La *coinbase transaction* è dove *nuovi* bitcoin vengono *creati* (*minati*, nel gergo di Bitcoin). Affinché la *coinbase transaction* dell'i-esimo blocco, per $i = 1, 2, \ldots$, sia valida, l'ammontare contenuto nell'output deve essere quello specificato dal protocollo per il blocco i-esimo, `new_btc`[i], più la somma delle *transaction fee* delle altre transazioni inserite nel blocco.

L'ammontare specificato dal protocollo per i nuovi bitcoin creati per ogni blocco parte da 50 e si dimezza ogni 210000 blocchi, $\texttt{new\_btc}[i] = 50/2^{\lfloor i/210000 \rfloor}$. In questo modo, il totale dei bitcoin in circolazione sarà sempre minore o uguale a 21 milioni.