# Blockchain e proof of work: come nasce l’oro del nuovo millennio?

Una cryptovaluta è un enorme registro di transazioni, un libro mastro dove vengono segnati i debiti e i crediti degli utenti che ne fanno parte.

A vuole inviare denaro digitale a B. Se il denaro è a conti fatti una stringa di bit, come si fa a rendere questa stringa non falsificabile, unica, e non duplicabile?

Una possibile soluzione consiste nel firmare una dichiarazione del tipo "A invia nX a B" con la chiave privata di A e, una volta che B ha controfirmato la dichiarazione, registrarla nel libro mastro. Come funziona una firma digitale?

Ogni utente possiede una coppia di chiavi: una pubblica e una privata. Una firma digitale, differentemente da una firma normale, non è un’entità fissa ma varia a seconda del documento firmato: il risultato della funzione che genera la firma dipende infatti sia dal documento da firmare, sia dalla chiave privata dell’utente che vuole firmare quel documento. Cambiare anche solo un bit del documento stravolge completamente la firma finale. La funzione di verifica prende invece in esame tre parametri: il messaggio, la firma del messaggio, e la chiave pubblica dell’utente che ha firmato il messaggio. Attraverso la funzione di verifica chiunque consulti il libro mastro può utilizzare le chiavi pubbliche di A e di B per verificare che le firme siano legittime e la transazione accettabile. Questa soluzione elimina il problema della creazione dal nulla di transazioni false, questa stringa può essere generata solo da A e controfirmata da B, ma non quello della duplicazione delle transazioni già esistenti (una volta creato il messaggio potrebbe essere copiato più volte sul libro mastro).

Per rendere una transazione unica si può assegnare ad essa un numero seriale univoco, ma per farlo c'è bisogno di un ente autorevole che assegni un numero di serie valido a ogni transazione. Questo ente dovrebbe occuparsi di controllare le transazioni, controllare i saldi di tutti gli utenti nel libro mastro, e fornire i numeri di serie univoci solo alle operazioni valide. In questo modo, una dichiarazione del tipo “A invia nX a B, codice seriale #123456” può essere registrata in maniera univoca e sicura sul libro mastro una volta che B, dopo essersi accertato con l’ente garante, decida di controfirmare la dichiarazione.

Sorgono due problemi: come fare a scegliere un ente autorevole? Fare in modo che i flussi di denaro vengano gestiti da un’entità centrale crea più vantaggi o svantaggi?

L’idea alla base della blockchain è quella di rendere tutti gli utenti del libro mastro l’ente di controllo stesso. In particolare, tutti gli utenti che utilizzano il libro mastro posseggono una copia del libro mastro stesso per controllare la validità delle transazioni. Questo libro mastro è appunto detto Blockchain.

All’interno della blockchain, quando A vuole inviare denaro a B, A firma il messaggio “A invia nX a B, codice seriale #123456” e invia questo messaggio a B; B può poi usare la sua copia della blockchain per controllare che la transazione sia legittima e, una volta confermata la validità della transazione, può controfirmarla per accettarla. Se la transazione viene accettata, B si occupa di trasmetterla a tutta la rete blockchain che viene riaggiornata con la nuova transazione, altrimenti la transazione viene ignorata e la blockchain non viene aggiornata.

Questo approccio però genera due problemi: 1) chi fornisce il codice seriale per la transazione? 2) come fare per impedire che A possa inviare la stessa dichiarazione, con lo stesso numero seriale, a più utenti diversi?

Caso ipotetico: A può inviare i messaggi “A invia nX a B, codice seriale #123456” a B e “A invia nX a C, codice seriale #123456” a C contemporaneamente; B e C controlleranno la validità della dichiarazione sulla loro blockchain che non è stata ancora aggiornata e, nel caso dovessero valutare la transazione nella stessa finestra d’aggiornamento temporale, entrambi controfirmeranno la transazione come valida e la trasmetteranno alla blockchain globale. Ma come fa la blockchain globale a decidere quale delle due transazioni con lo stesso numero seriale è valida? E anche se dovesse decidere, il codice seriale è univoco, quindi una delle transazioni verrebbe scartata.

Per evitare questa situazione l’utente che deve controfirmare una transazione non dovrebbe limitarsi a verificare la transazione da solo. Quando A invia il messaggio “A invia nX a B, codice seriale #123456”, B dovrebbe come prima cosa controllare la validità sulla sua copia della blockchain, poi dovrebbe ritrasmettere il messaggio a tutta la rete e infine, una volta che la transazione è stata validato dalla rete, controfirmarla e ritrasmetterla a tutto il network. In questo modo, una situazione come quella precedente in cui A invia due pagamenti con lo stesso codice seriale a due persone diverse non potrebbe verificarsi, in quanto la rete stessa segnalerebbe ai due utenti che c’è un problema con la transazione prima che questi possano accettarla.

Come decidere però quanti utenti devono accettare una transazione prima che essa sia definita valida? Non possono essere tutti perché non è possibile conoscere a priori il numero di utenti della rete e, sempre per lo stesso motivo, non è possibile definire una porzione o comunque un numero fisso di utenti necessari per l’approvazione. Un possibile problema, inoltre, è che un utente potrebbe prendere il controllo del sistema di decisione creando un numero enorme di utenti fittizi che prendano il controllo della rete ogni volta che bisogna validare una transazione. È dunque necessario un sistema di consenso che non possa essere aggirato.

La rete blockchain utilizza un algoritmo di consenso basato sul protocollo *proof of work.* Il protocollo *proof of work* prevede la combinazione di due idee: 1) rendere artificiosamente costosa da un punto di vista computazionale la validazione delle transazioni e 2) premiare gli utenti che aiutano a validare le transazioni. Il vantaggio del rendere la validazione di una transazione complessa è che in questo modo non può essere influenzata dal numero di identità che un utente controlla all’interno della rete, bensì dalla potenza computazionale che può utilizzare per influenzare l’esito della verifica. È possibile, con alcune accortezze, fare in modo che sia impossibile (o comunque *estremamente* sconveniente) avere la potenza computazionale necessaria per soverchiare il resto della rete.

La *proof of work* prevede il seguente scenario: A trasmette alla rete il messaggio “A invia nX a B, codice seriale #123456”; man mano che gli utenti della rete ricevono il messaggio lo aggiungono ad una coda di messaggi simili, ancora non approvati, che hanno ricevuto fino a quel momento; gli utenti che vogliono convalidare le transazioni in coda devono prima risolvere un puzzle complesso a livello di calcolo, la proof of work, perché la loro verifica possa essere considerata valida ed essere così trasmessa alla rete.

Il puzzle da risolvere consiste nel trovare, dato un blocco di transazioni e una funzione crittografica SHA-256, un numero x che, dato in input alla funzione insieme al blocco di transazioni, generi un hash che cominci con un numero di 0 consecutivi deciso in partenza. La difficoltà del puzzle dipende dal numero di 0 consecutivi che si devono trovare. La peculiarità che rende la risoluzione di questo puzzle difficile è che l’output di una funzione hash crittografica si comporta a tutti gli effetti come un numero casuale: cambiare anche solo di 1bit l’input della funzione stravolge l’output finale in maniera imprevedibile. Non appena un utente trova un numero che soddisfi i requisiti lo trasmette alla rete insieme al blocco di transazioni da approvare. Una volta verificata la legittimità della soluzione, tutti gli utenti della rete aggiornano la blockchain con l’ultimo blocco di transazioni.

Dal momento che l’unico modo per trovare un numero che soddisfi i requisiti consiste nell’andare per tentativi, e che la complessità dell’operazione è estremamente elevata (può essere manipolata in maniera arbitraria, ad esempio nel caso della rete bitcoin che è tarata per risolvere in media un blocco ogni 10 minuti), la potenza di calcolo necessaria per trovare quel numero è molto alta. È dunque necessario trovare un incentivo perché la gente sia spinta a usare potenza computazionale per validare questi blocchi di transazioni.

Il processo di validazione nella blockchain è chiamato mining. Per ogni blocco di transazioni validato, il primo miner che riesce a validarlo riceve come ricompensa una determinata quantità di credito, nuova valuta che viene generata all’interno della blockchain. La proof of work è insomma una sorta di gara a chi è più veloce ad approvare nuovi blocchi di transazioni: le chance di vincere il blocco dipendono dalla percentuale di potenza computazionale che può impiegare per risolvere il puzzle. Ad esempio, se un miner controlla l’1% della potenza globale della rete di validazione, la probabilità che vinca un blocco è di circa l’1%.

Un blocco è considerato valido solo se ha una proof of work valida, e per fare in modo che la blockchain sia ordinata in maniera cronologica ogni blocco contiene l’hash del blocco precedente. Dal momento che gli utenti della blockchain prendono come riferimento la catena con la maggior potenza computazionale spesa per generarla, quando un utente della rete riceve due blockchain diverse tra di loro, l’utente sceglierà quella con la catena più lunga; in caso di pareggio, aspetterà fino a quando una delle due non diventerà n blocchi più lunga dell’altra. In un sistema del genere, validare blocchi contenenti transazioni illegittime è estremamente complesso. Se A invia a B un blocco non legittimo, B non lo accetterà subito perché allo stesso momento riceverà i blocchi legittimi trasmessi dagli altri miner; considerando che l’utente B sceglierà sempre la catena più lunga, per riuscire nel suo intento A dovrebbe continuare a generare da solo più blocchi rispetto a tutta la rete di miner. Per poter riuscire nel suo intento, A da solo dovrebbe avere più del 50% della potenza di calcolo dell'intera rete.