Le blockchain

Vincenzo Scotti M63/693

Universita' degli Studi di Napoli Federico II

September 25, 2017

Introduzione

- Tecnologia emergente per l'implementazione di basi di dati distribuite
- Particolarmente adatta a sistemi transazionali
- Elevata affidabilita' essendo la base di dati replicata su ogni nodo
- Robusta rispetto alla presenza di *nodi malevoli*, fin tanto essi sono in minoranza

Introduzione

- Tecnologia emergente per l'implementazione di basi di dati distribuite
- Particolarmente adatta a sistemi transazionali
- Elevata affidabilita' essendo la base di dati replicata su ogni nodo
- Robusta rispetto alla presenza di *nodi malevoli*, fin tanto essi sono in minoranza

Per questo motivo stanno nascendo le piu' svariate applicazioni

- criptomonete (bitcoin, litecoin, ethereum, ...)
- social network (steem, ...)
- contratti
- sistemi di gestione documentale
- ed altro...

Struttura dei dati

- Struttura dati formata da una lista linkata di blocchi, immaginabili come *transazioni* che alterano lo stato del sistema
- Un blocco mantiene informazioni sulla transazione nel dominio di business, oltre ad altri campi necessari a garantire la sicurezza del sistema

Le blockchain come sistema distribuito

- L'intera struttura dati e' replicata su ogni nodo => no single point of failure
- Le transazioni sono memorizzate in chiaro e verificate da ogni nodo della rete
- La blockchain principale e' quindi la catena di blocchi sulla quale la "maggioranza" dei nodi concorda

Le blockchain come sistema distribuito

- L'intera struttura dati e' replicata su ogni nodo => no single point of failure
- Le transazioni sono memorizzate in chiaro e verificate da ogni nodo della rete
- La blockchain principale e' quindi la catena di blocchi sulla quale la "maggioranza" dei nodi concorda

Se pero' la blockchain e' di dominio pubblico, servono meccanismi per:

- l'*autenticazione* e l'*autorizzazione* dei nodi
- il controllo dell'*integrita'* di tutta la catena (anche 500k+ blocchi)

Utilizzo degli hash crittografici

I sistemi blockchain fanno uso di hash crittografici per:

- generazione di un ID univoco per blocco
- controllo d'integrita'
- collegamento dei blocchi tra loro
- realizzazione del *proof of work* (spiegato dopo)

Utilizzo della cifratura asimmetrica

- Non c'e' un database delle credenziali (sarebbe SPOF)
- I nodi sono identificati da una coppia di chiavi asimmetriche => ogni nodo collegato alla rete puo' generare una coppia di chiavi e partecipare allo scambio di messaggi
- Le transazioni (solo i dati di business) sono firmati digitalmente dal nodo autore

Tutti i nodi della rete, nel validare il blocco, controlleranno se "*il nodo N (identificato dalla coppia di chiavi X/Y) e' autorizzato ad emettere la transazione T*"

Struttura del blocco

- Hash del blocco
- Chiave pubblica dell'autore
- Informazioni di business
- Firma dell'autore
- Hash del blocco precedente
- Proof of work (vedi dopo)

Scenari d'attacco

Chiamiamo 'M' un generico nodo malevolo

- M prova a modificare/corrompere una transazione creata da A, ne ricalcola l'hash e lo invia sulla rete

Scenari d'attacco

Chiamiamo 'M' un generico nodo malevolo

- M prova a modificare/corrompere una transazione creata da A, ne ricalcola l'hash e lo invia sulla rete

Attenzione! La firma non sarebbe piu' valida, quindi il blocco non sarebbe accettato dai nodi buoni

Scenari d'attacco

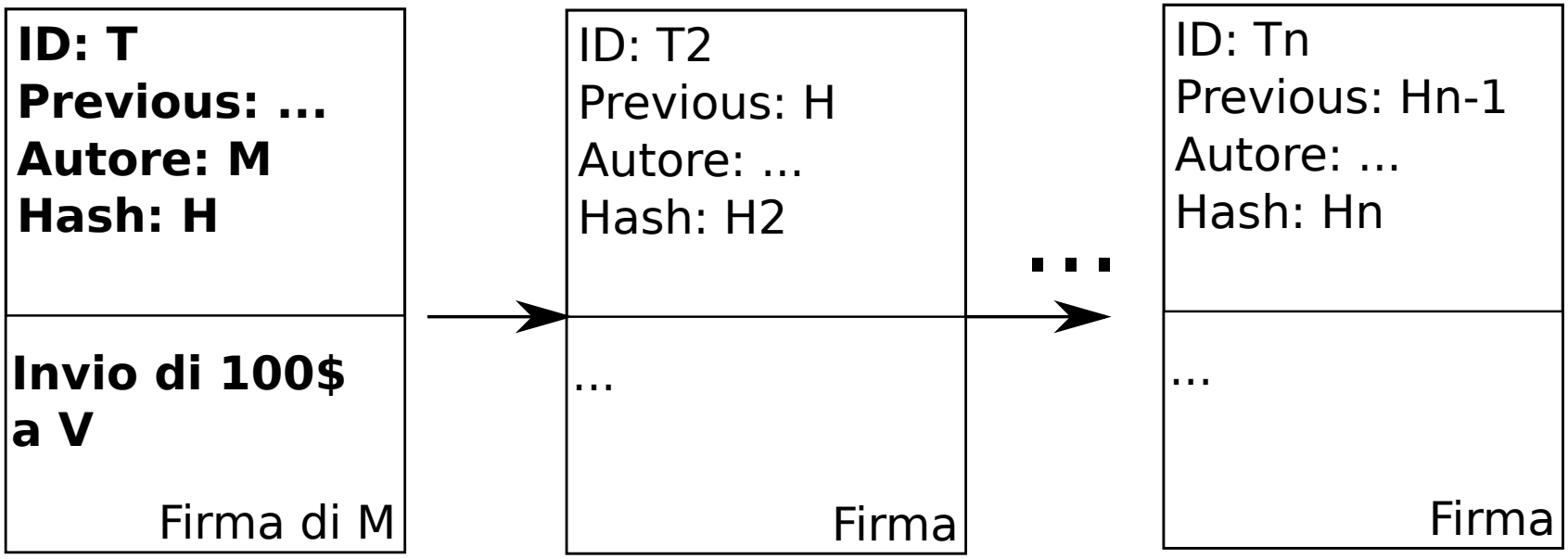
Chiamiamo 'M' un generico nodo malevolo

- M prova a modificare/corrompere una transazione creata da A, ne ricalcola l'hash e lo invia sulla rete

Attenzione! La firma non sarebbe piu' valida, quindi il blocco non sarebbe accettato dai nodi buoni
Le uniche transazioni modificabili da M sono quindi quelle che egli stesso puo' rifirmare, ovvero quelli di cui e' autore

Il "double spending problem"

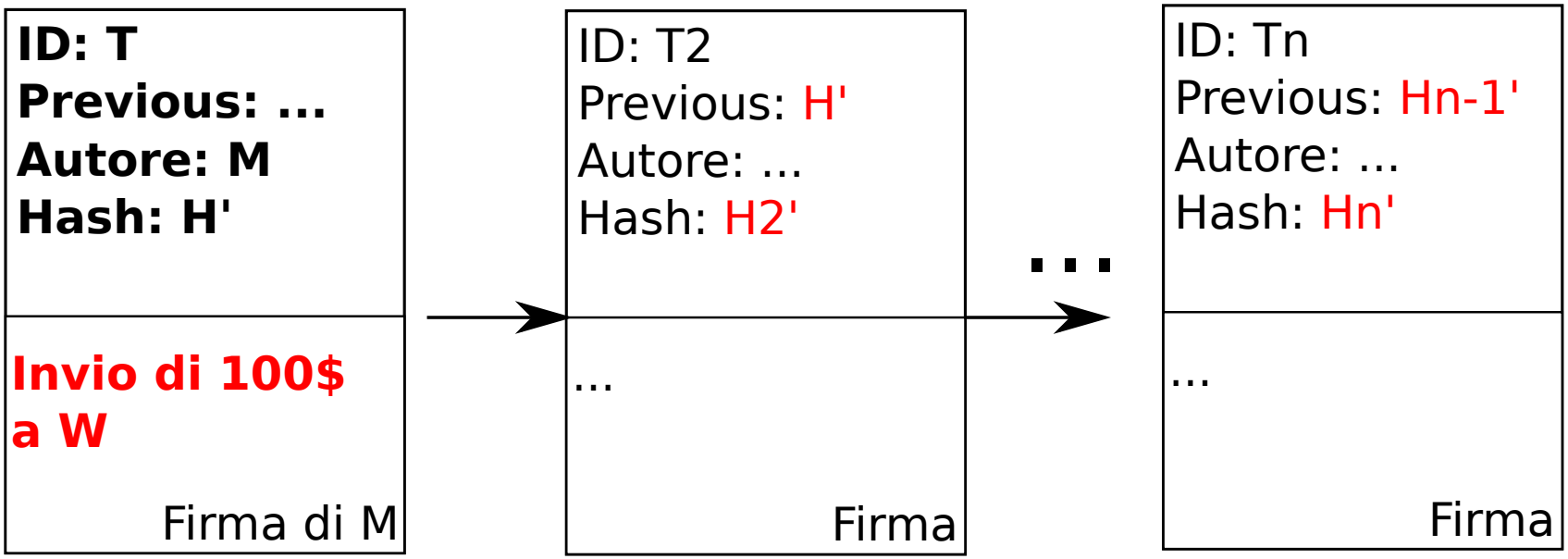
- M acquista online un oggetto da V
- V aspetta un pagamento da M, per effettuare la spedizione



- V trova il pagamento nella blockchain, e procede alla spedizione

Il "double spending problem" (2)

- M altera la propria transazione, e tutte le successive, in modo da reindirizzare il pagamento ad un altro nodo



- Il problema e' che nulla impedisce ad un nodo malevolo di modificare una propria transazione passata, e aggiornare i puntatori dei successivi blocchi

Il proof of work

- La **vera** novita' delle blockchain e' stata l'introduzione del proof of work
- Il proof of work e' un lavoro computazionale da eseguire in fase di convalida del blocco
- Ha una natura probabilistica e complessita' media molto elevata
- Ma deve essere facile verificarne l'esecuzione
- I nodi in parallelo "assolvono" al proof of work (in un processo chiamato comunemente **mining**)
- Di solito c'e' una ricompensa per chi lo completa prima, quindi il mining diventa una vera e propria gara

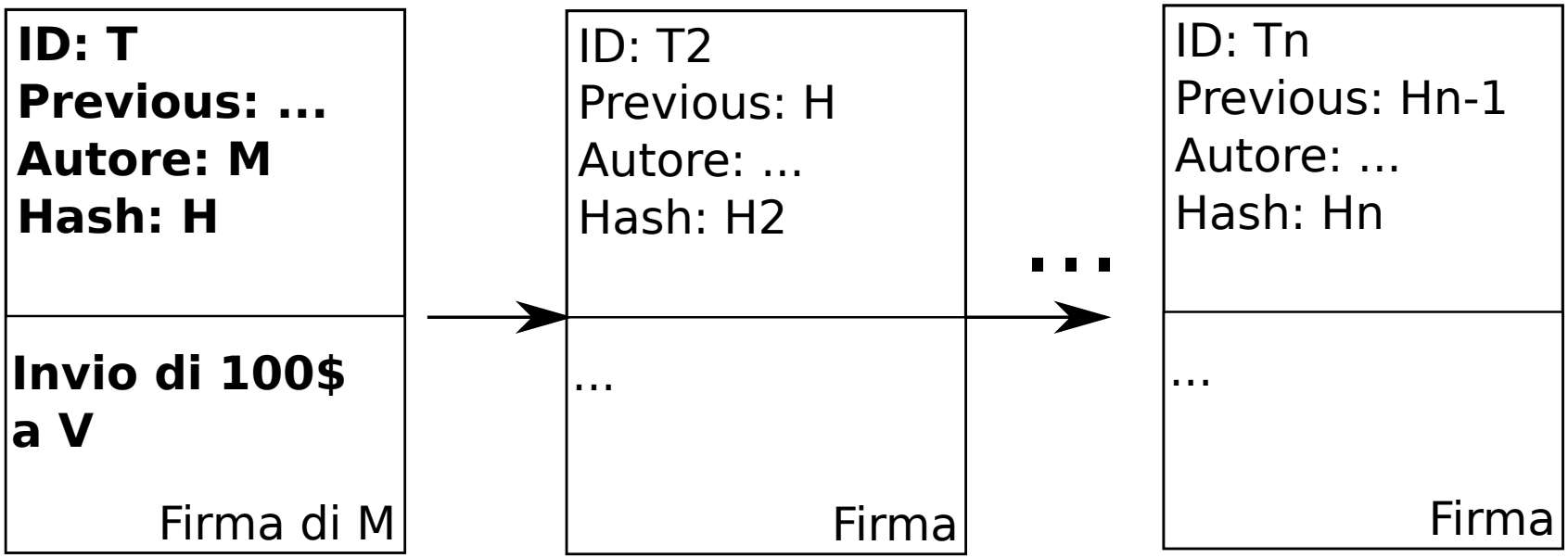
POW e hash crittografici

Trovare un nonce da aggiungere al blocco affinche' l'hash corrispondente sia inferiore ad una soglia S

- La complessita' e' elevata poiche' lo spazio dei campioni e' enorme
- Se la funzione di hash e' accettabile, i campioni hanno tutti la stessa probabilita' di essere quello "giusto"
- La durata dell'operazione e' quindi casuale, e con valore medio pari alla meta' dei campioni da testare
- D'altro canto, la verifica della correttezza del lavoro e' immediata: basta calcolare l'hash del blocco (incluso il nonce) e controllare che esso soddisfi la condizione richiesta

POW come soluzione al "double spending problem"

- Riprendiamo lo scenario precedente, supponendo che M voglia modificare una sua vecchia transazione



- Modificando T in T', l'hash andra' a cambiare, cosi' come per tutte le transazioni seguenti
- Vanno quindi ricalcolati tutti i POW dei blocchi a seguire

POW come soluzione al "double spending problem" (2)

- Inoltre durante questo processo, nuovi blocchi potrebbero essere emessi ed approvati dal resto della rete
- Diventa quindi una **corsa** tra il nodo M, che deve recuperare lo svantaggio, e il resto della rete
- **Se la potenza computazionale del nodo M e' minore della potenza del resto della rete, la probabilita' di riuscita decade esponenzialmente col numero di blocchi da recuperare**
- Una transazione in una blockchain e' quindi considerata **sicura** se viene superata da un numero sufficiente di blocchi

Bitcoin

- E' la prima implementazione di moneta virtuale basata su blockchain
- ~500k blocchi (ad oggi)
- SHA-256 come funzione di hashing
- ECDSA per firmare le transazioni
- Piu' transazioni per blocco
- Nonce di 4 byte e soglia S adattativa in modo da mantenere costante la frequenza di emissione dei blocchi (~10 minuti)
- I miner devono quindi provare (tutte le combinazioni di nonce) X (tutte le combinazioni di transazione per blocco), fino ad ottenere un hash valido
- Una transazione e' considerata sicura se essa e' seguita da 6 blocchi

BCSim

- Simulatore di blockchain realizzato con tecnologie Java (Swing, JCA, ...)
- Le transazioni sono dei messaggi pubblici (analogo ad un social network)
- SHA-1 come funzione di hashing
- SHA-1 + RSA per firmare le transazioni
- Timeout casuale (in range configurabile) per simulare POW e ritardi della rete di interconnessione