**Tecnologie di blockchain**

**lezione 1**

strutturazione argomenti, lezioni, materiale, esame finale

**Introduzione della blockchain**

**BlockChain(**catena di blocchi**):** è un **Registro condiviso e immutabile**

**Registro:** archivio di info memorizzate con un ordine temporale

le info della blockchain sono conservati in **blocchi** di dimensioni definite (BTC è 1MB)

ogni blocco viene memorizzato con una data

(*unix timestamp*, un numero intero in secondi, parte dal 1/1/1970)

il registro è condiviso **pubblicamente** tra tutti i nodi della rete blockchain

ogni **nodo** detiene la stessa copia del registro

i nodi sono collegati con una rete **peer-to-peer** (sono allo stesso lvl)

il registro non può essere modificato una volta memorizzata un blocco  
i blocchi sono collegati tra loro tramite **funzioni hash**(restituisce un codice digest di 24 caratteri esadecimali, minima modifica=cambia di nuovo)  
la funzione **non è invertibile**

**Catene di blocchi:**

*meccanica*

blocco genesi (serve per avviare procedura transazioni)

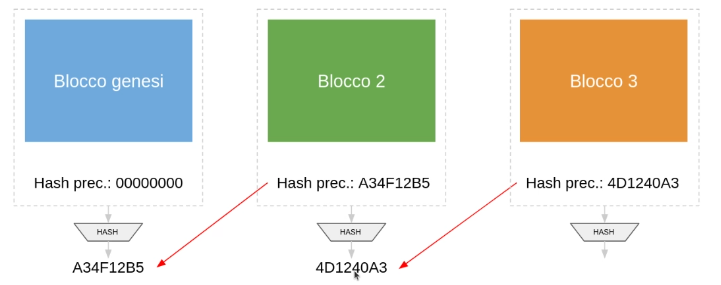
V

ogni blocco contiene hash di quello precedente (per genesi è 0)

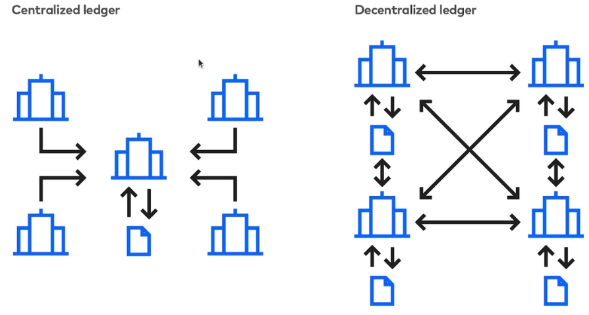
V

si raccoglie info blocco e codice hash precedente, per generare funzione hash

V

hash del blocco sarà riferimento per blocco successivo  
  
*perché serve hash del blocco precedente?* serve a legare i nuovi blocchi a quelli precedenti, ovvero il senso della chain

**Registro decentralizzato**

****

*sistema centralizzato = architettura client-server*

*sistema decentralizzato = computer che comunicano tra loro (no server terzi o altro)*

*fondamentale per il mantenimento della rete è la trasparenza/condotta utenti*

**Rete Peer-To-Peer**

protocollo in cui ogni nodo collegabile tra vicini(distribuzione casuale), per passare documenti, distribuzione spontanea/cascata: quando qualcuno riceve un file tutti lo riceveranno

nb: 455 GB dimensione blockchain di BTC

**Gossip  
algoritmo P2P,** modo + rapido di diffondere le informazioni

funzionamento: n nodi, calcolo distribuzione: log(n)

https://ctufaro.github.io/GossipPlot/index.html

**Bitcoin Satoshi nakamoto article**

documento che introduce argomenti precedenti

risolve problemi di **double spending** su una rete decentralizzata con il **Proof Of Work (PoW)**, impedisce di copiare monete e riutilizzarle

prima di questo si provava a risolvere con reti centralizzate

tentativi di monete digitali: DigiCash B-money e bitgold

**CypherPunk:** movimento intellettuale anni ‘80 per creare un sistema per proteggere la privacy.

**Bitcoin:** prima criptovaluta 2008

1 BTC = 70.415,07 euro il 8/11/2024

market cap 1.50 trilioni di dollari

**Come Funziona btc?**

ogni persona ha uno o più indirizzi, sono anonimi

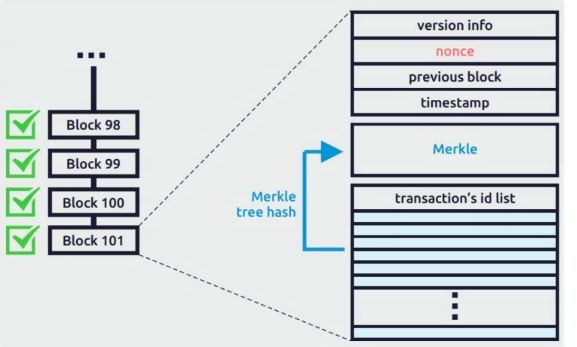
i bitcoin di trasferiscono da un indirizzo ad un altro (**transazione**)

le transazioni sono pubbliche, quindi tipo semi anonime   
 (indirizzo A ha trasferito tot btc a indirizzo B)  
  
**Come si generano btc?**

sono generati dai **miner:** nodi di rete che convalidano creazione di blocchi ogni 10min (risolvono problema, quando risolto ricevono i btc a velocità fissa)

numero massimo di btc in circolazione 21 milioni

**Struttura Blocco**

****

**Hash**  
funzione che collega le catene di dati tra loro

non è invertibile, non posso ottenere matematicamente il valore di input a lvl pratico computazionalmente impossibile  
modello codifica hash +noto SHA256

***proprietà:***

rapidità e facilità calcolare hash messaggio

non invertibilità

resistenza debole alle collisioni   
 (possibilità che un hash sia uguale ad un altro, perché i codici hash sono finiti)

resistenza forte alle collisioni

(difficile trovare messaggio con lo stesso hash)

**SHA256**

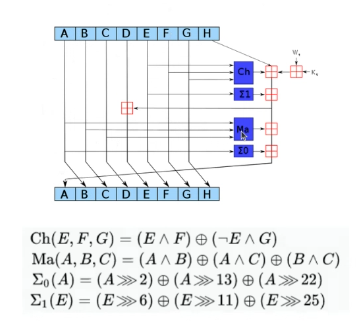
messaggio diviso in blocchi di 512 bit

V

iterati con una funzione di compressione su insieme di 8 valori da 32 bit  
(come si mischiano i bit nei blocchi)

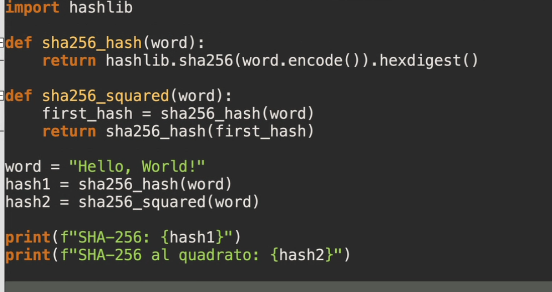
V

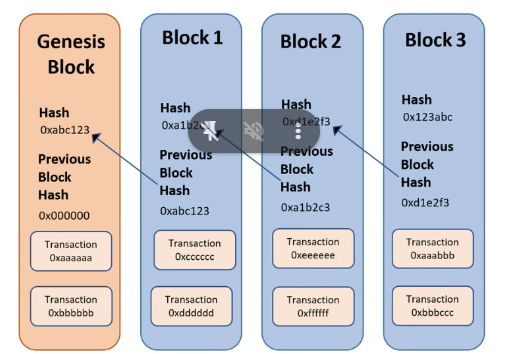
risultato : concatenazione di blocchi ( 8 x 32 = 256 bit)

****

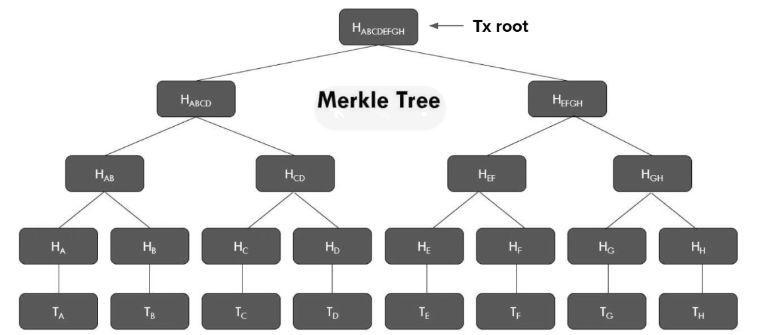
BTC usa **SHA256^2** (2 volte sha256)

altri codici esercizio provare con testo



**Struttura catena**

**Perché non conviene fare l’hash per ogni transazione?** altrimenti lo spazio occupato sarebbe troppo

si risolve il problema per N transazioni con il **merkle tree  
**

si legge dal basso verso l'alto (ne resta 1)

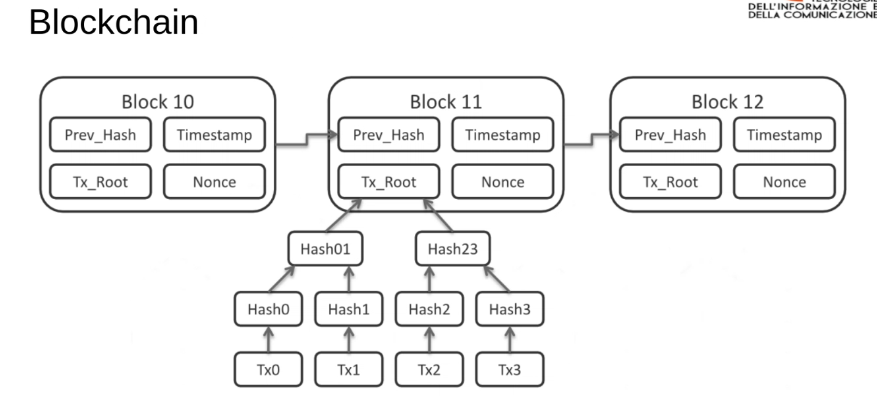
T = transazione

H = hash transazione

è un algoritmo ricorsivo

ragioni della meccanica: per riconoscere l’hash di una transazione è + facile raggiungere l’hash di una transazione sola

quindi la **BLOCKCHAIN** struttura completa



nb la blockchain aumenta sempre di dimensione(il giusto )

