**Lezione 13/05/2024**

**Concetti di economia e finanza decentralizzata**

**blockchain e robetta varia**

**Blockchain:** registro di transazioni condiviso e immutabile, formata da una catena di blocchi che si connettono tra loro, decentralizzato….

***registro:*** archivio per memorizzare informazioni in ordine temporale (file)

le info (come se fosse uno cartaceo) in una bc hanno dimensioni prestabilite (1MB max ecc...)

nelle blockchain non vanno info corpose

***condiviso:*** *tutti i partecipanti hanno una copia dello stesso registro*

quindi se qualcuno altera il proprio registro e viene sottoposte a confronto con le altre copie si nota la scorrettezza viene buttato fuori (basta che prevalga la maggioranza o il 50% di un gruppo): questa rivoluzionaria precauzione per sicurezza comporta allo rispettare una condotta corretta e si viene ricompensati

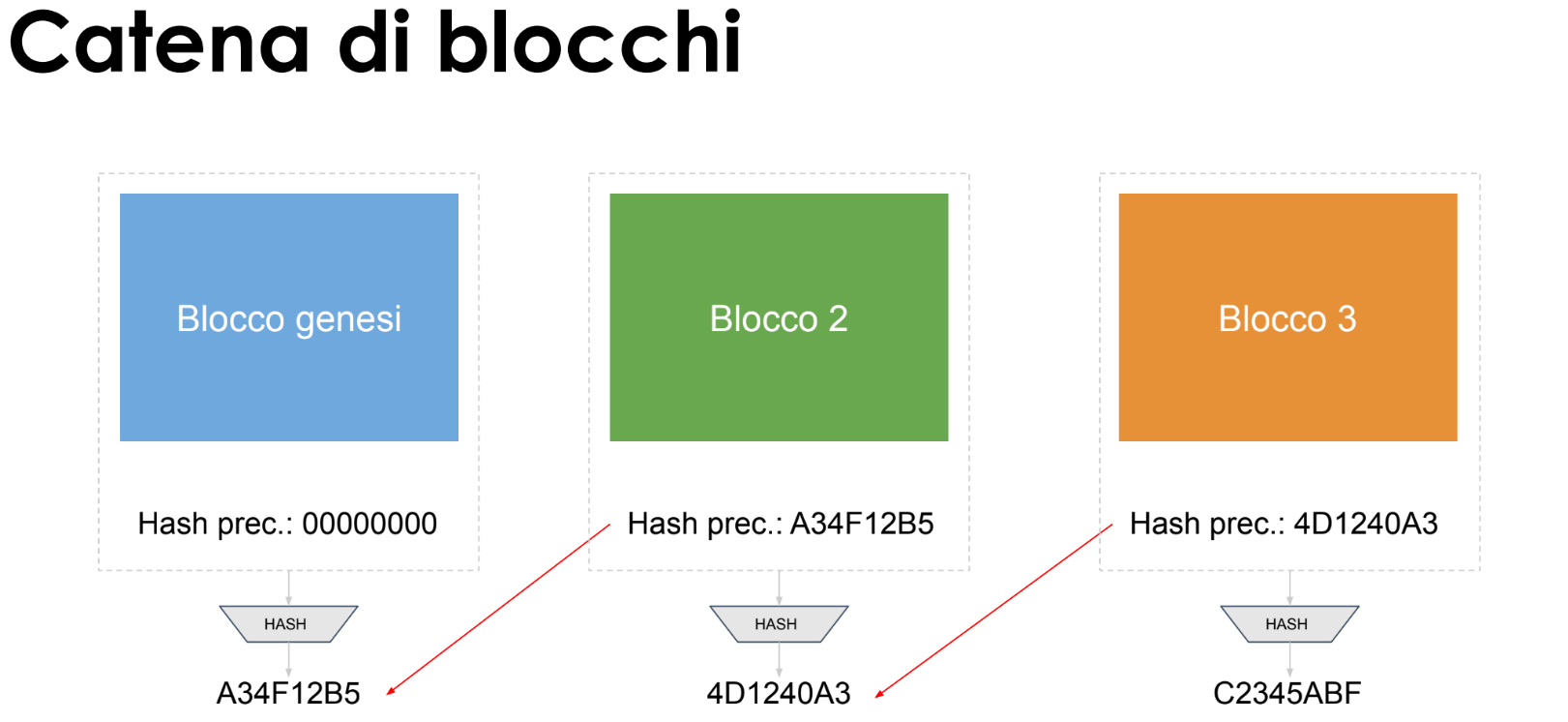
la rete dei nodi è **peer-to-peer : P2P** , tutti possono quindi condividere info tra di loro

***immutabilità:*** memorizzato il dato non puoi modificarlo

i blocchi sono collegati tra loro con **funzioni hash:** dai un input specifico ti da un output(?)

es: se carico un contratto a un'ora precisa su una blockchain pubblica otterrò un codice hash

a lvl giuridico lo si può verificare il codice per certificare l’originale doc che conferma l’azione



il *blocco genesi* è il blocco iniziale che non ha nulla inizialmente, poi si fa l’hash e si ottiene il suo codice.

poi generando il secondo blocco con le sue info deve essere legato al blocco precedente(genesi) e poi calcolare il codice hash del secondo blocco legato al precedente e così via.

ogni nuovo blocco è legato a quello precedente

è un processo lungo, però in questo modo la fiducia è intrinseca al sistema matematico e non alle persone

nell’avere un sistema decentra/centralizzato, bisogna considerare e garantire una

**scalabilità**: stabilità del sistema in base alla qtà di utenti che fanno uso del sistema

**Gossip:** [**https://ctufaro.github.io/GossipPlot/index.html**](https://ctufaro.github.io/GossipPlot/index.html)

algoritmo usato per far si che un nodo inva info ad un altro nodo e si diffonda ecc

**Origine blockchain e BTC:** 2008, Satoshi Nakamoto(pseudonimo), check pdf / bibbia bitcoin

non è un articolo scientifico (informatico)

(considerazione di come vengono revisionate e analizzate le ricerche scientifiche)

con bitcoin e allegate si risolve ***problema del double-spending*** con il ***proof of work (PoW):*** pensato sull’ avidità ed egoismo umano

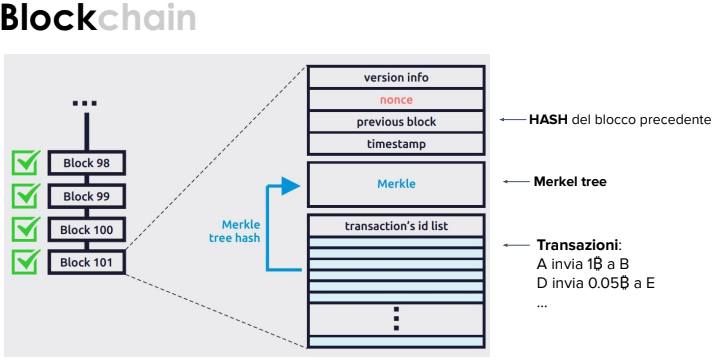
“antenati falliti” di bitcoin, per connettere nodi(dash nel 1991), o valuta digitale(falliti)

**cypherpunk:** principi di crittografia come liberazione per la protezione della privacy

**BITCOIN: prima criptovaluta al mondo**

per lavorare con bitcoin e blockchain si fa uso di **indirizzi** anonimi(gestito da utente o servizio); ciò permette le **transazioni** (sono pubbliche perché scritte sul registro, ogni spostamento è tracciabile)

i **miner** sono i nodi che creano i btc ogni 10 minuti costanti, crescita lineare, max 21 mln BTC in circolazione, sistema scalabile.



la **Funzione hash** è una funzione non invertibile che mappa una stringa di

lunghezza arbitraria in una stringa di lunghezza predefinita(lunga quanto vuoi es. 32 bytes)

praticamente prende in input un file e restituisce un output con una lunghezza predefinita in un formato esadecimale

Esempio: **SHA256(“Hello World”) =** funzioneHash(input)

a591a6d40bf420404a011733cfb7b190d62c65bf0bcda32b57b277d9ad9f146e (output codice)

**formati di codifica**

considerazione dei formati (binario esadecimale…)

<https://www.digikey.it/it/resources/conversion-calculators/conversion-calculator-number-conversion>

**numero binario:** sequenza di 0 e 1(5 volt) (sono bit, cmq il computer “conta” fino a due numeri)(check calcolo)

es: 10111 = 23 (base 2)

in python

num=**0b**num che converte

**bin(**num**)** è inverso

num.**bit\_length()** misura bit del numero intero

**numeri ottali:** 0 1 2 3 4 5 6 7

es 23 = 27 in ottale (base 8)

in python

num=**0o**num che converte

**oct(**num**)** è inverso

**numero esadecimale:** uso 16 simboli (0 1 2 3 4 5 6 7 8 9 a b c d e f)

es 23 = 17 in esadecimale (base 16)

in python

num=**0x**num che converte

**hex(**num**)** è inverso

in un byte (8 bit) posso scrivere **2^8** numeri = 256

ASCII, blink, caratteri, codici di controllo…

da 0 a 9 sono da 48 a 57

da a a z sono da ! a !

da A a Z sono da ! a !

in python

num = **b**’abc’ converte caratteri in byte

num.**to\_bytes()** inverso e specifica il numero di bytes

chr(num) da quel numero ad ascii

**encode:** converte stringhe in byte

es: test.encode(“utf-8”)

**Base64:** sistema di codifica (non cifratura) molto usato che ha un formato composto da 64 caratteri(alfabeto, numeri, caratteri base) (meno simboli molti non leggibili)

es codifica Base64 da

Man = TWFu

Convertitore: <https://www.base64decode.org/>

in python (import Base64 e check resto)

dal codice hash, per risalire all’indirizzo che ha codificato un doc in funzione hash, si impiega tantissimo livello usando algoritmi pure

prova concetto di non invertibilità, computazionalmente impossibile non convertire

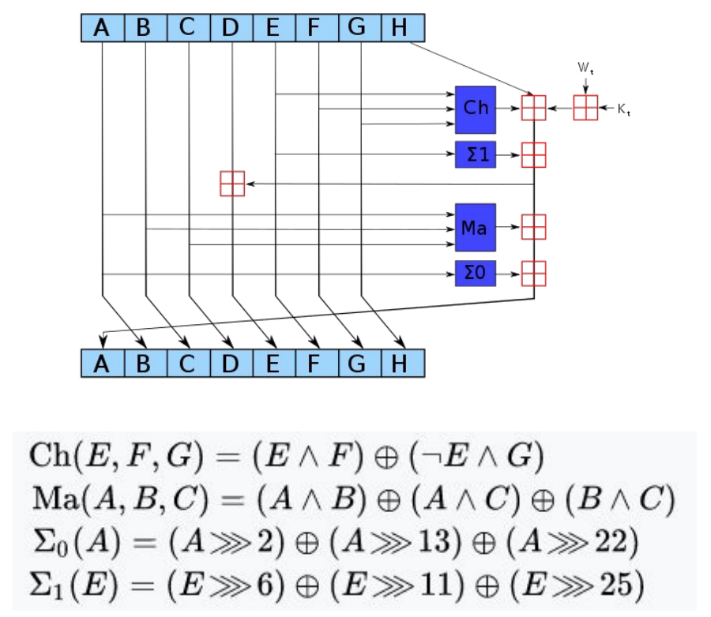
inoltre È computazionalmente difficile modificare un messaggio senza modificare il relativo hash (resistenza debole alle collisioni: ci possono essere + docs con lo stesso hash)

Dato un messaggio è computazionalmente difficile trovarne un altro con lo stesso hash (resistenza forte alle collisioni)

funzione hash: **SHA256**

suddivide msg in **blocchi da 512 bit**

*operazioni computazionali (lavora con i bit grazie operatori AND NOT XOR ..)*



btc lo usa ma lo fa due volte **SHA256^2**