Scuola di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica

Tesi di Laurea

UNIBLOCK: BLOCKCHAIN UNIVERSITARIA

UNIBLOCK: UNIVERSITY BLOCKCHAIN

CLAUDIO NAVE

Relatore: *Francesco Tiezzi*

Anno Accademico 2021-2022

INDICE

1

INTRODUZIONE

Capitolo di introduzione. Fatto alla fine.

2

NOZIONI DI BASE

2.1 BLOCKCHAIN

Nella sua più primitiva definizione una blockchain è una struttura dati in grado di registrare informazioni e garantirne l'immutabilità nel tempo, ridondando i dati su un gran numero di nodi senza che sia necessaria un'entità centrale che vigili su possibili minacce o malfunzionamenti. Concepita nel 2008 dallo pseudonimo Satoshi Nakamoto, la cui vera identità è tutt'oggi sconosciuta, [7] come mezzo per decentralizzare il mondo delle transazioni finanziarie, è stata oggetto nel corso del tempo di un forte interesse accademico e commerciale venendo applicata in un gran numero di scenari diversi. Pur essendo presenti sul panorama attuale svariati esempi di blockchain anche molto diverse tra loro in complessità e implementazione, possiamo individuarne alcune concetti di base comuni a tutte loro.

2.1.1 *Struttura a blocchi*

Ogni blockchain è composta da vari blocchi di informazione legati fra loro tramite l'uso di hash crittografici (sez. 2.3.1). Infatti per garantire l'integrità dei dati ogni blocco contiene l'hash del blocco precedente, così che un cambiamento nell'informazione contenuta in un blocco verrà subito riconosciuto da un semplice controllo sugli hash. All'interno di un blocco le informazioni sono generalmente organizzate in una lista di transazioni, chiamate così a causa del loro concepimento in ambiente di transazioni economiche nella rete Bitcoin.

Per ottimizzare il processo di hashing delle transazioni viene generalmente impiegato un albero di Merkle: si costruisce un albero binario avente come foglie gli hash di tutte le transazioni di un blocco e ogni altro nodo conterrà l'hash della concatenatione dei suoi due figli. Procedendo

così dal basso verso l'alto l'hash contenuto nella radice dell'albero sarà l'hash che rappresentetà l'intera lista di transazioni e l'unico elemento che verrà incluso nel calcolo dell'hash del blocco. Tale struttura ad albero permette una notevole flessibilità nel processo di verica delle transazioni, infatti se si è interessati alla verifica dell'integrità di una singola transazione basterà chiedere agli altri nodi di fornire gli hash intermedi utili a ricostruire solo il particolare ramo della transazione in esame. Del resto una modifica a una transazione qualsiasi comporterebbe una modifica in cascata di tutti gli hash fino a quello di radice, venendo bloccata subito dal controllo degli hash sui blocchi.

2.1.2 *Modalità di accesso alla rete*

Una prima differenza fondamentale delle blockchain è la modalità con cui i nuovi nodi possono entrare a far parte della rete. Nelle blockchain pubbliche qualunque dispositivo può diventare un nodo della rete senza alcun controllo sulla sua legittimità. Per questo motivo tali blockchain prendono il nome di permissionless blockchain in quanto non c'è alcuna differenza gerarchica tra i vari nodi e chiunque può entrare a farne parte. In generale questo tipo di blockchain presenta un numero di partecipanti molto elevato grazie alla bassa soglia di entrata. Nelle blockchain private invece l'entrata nella rete è preceduta da una fase di autenticazione del soggetto come l'appartenenza a una determinata azienda. Tali blockchain sono quindi confinate nelle realtà che le sviluppano e le loro pool di nodi sono di conseguenza molto ridotte in quanto solo chi è qualificato può entrare a farne parte. Inoltre riflettendo la gerarchia dell'organizzazione proprietaria della blockchain sono in genere presenti differenze di responsabilità tra i vari nodi, da qui il nome di permissioned blockchain. Possiamo identificare inoltre un terzo tipo di blockchain, quello ibrido, in cui vengono uniti alcune caratteristiche delle blockchain pubbliche con quelle private. Nelle blockchain ibride alcune funzioni sono lasciate come pubbliche mentre altre richiedono una preventiva autenticazione. Le blockchain pubbliche sono di gran lunga il tipo più comune di blockchain, ma quelle private stanno guadagnando terreno avendo attirato l'attenzione del mondo finanziario [6].

2.2 ALGORITMO DI CONSENSO

Data la natura distribuita di una blockchain è di fondamentale importanza la ricerca del consenso tra i nodi, ovvero che ogni transazione generata venga validata e diffusa attraverso la rete rimanendo inalterata. Il primo algoritmo di consenso concepito è la proof of work: l'immutabilità dei dati è garantita dalla difficoltà di computare rapidamente un puzzle crittografico. Nella rete Bitcoin ad esempio un nuovo blocco è considerato valido quando viene trovato un numero tale che inserito nell'header del blocco stesso rende il suo hash inferiore a una certà quantità. Variando tale quantità è possibile modulare il carico dei blocchi generati dall'intera rete, dando il tempo ai blocchi di diffondersi tra i vari nodi. Ogni nodo dà la sua fiducia alla catena di blocchi in cui è stata spesa la maggior quantità di lavoro computazionale. Un algoritmo di consenso concepito più recentemente è la proof of stake: i nuovi blocchi vengono validati da nodi scelti casualmente in base a quanta valuta hanno investito nella rete. Pur ritenendo un certo livello di casualità, l'algoritmo di selezione del prossimo validatore privilegia i maggiori scommettitori. La legittimità della rete è garantita perciò dal fatto che chi ha investito maggiormente avrà interesse nel suo corretto funzionamento. La rete Ethereum prevede di effettuare il cambio da proof of work a proof of stake nei prossimi anni. Per un'analisi dettagliata sugli algoritmi di consenso fare riferimento a [3].

2.2.1 *Privacy e sicurezza*

Nelle blockchain pubbliche il contenuto dei blocchi è disponibile a ogni nodo senza alcuna protezione. Assume grande importanza quindi avere sempre ben presente quali informazioni si stanno immettendo nella rete in chiaro e quali invece si cerca di proteggere. Nella rete Bitcoin tutto il contenuto di un blocco è presente completamente in chiaro tanto da poter ricostruire la storia di ogni singolo bitcoin fino al momento della sua coniazione. In questo caso la privacy offerta agli utenti si limita alla loro anonimizzazione nella rete offrendo indirizzi usa e getta senza legami con l'identità legale del soggetto che li possiede. Anche la rete Ethereum non prevede di base alcuna forma di cifrazione del contenuto dei blocchi, essendo tuttavia presenti delle forme di privacy a zero conoscienza implementabili come zk-SNARKS per nascondere integralmente il contenuto di una transazione [5]. Discorso diverso invece per le blockchain private in quanto la preventiva autenticazione dei nodi permette di garantire che

le informazioni contenute nei blocchi siano consultabili solo da soggetti autorizzati, non rendendo necessari complessi sistemi di crittografia per garantire la riservatezza delle informazioni.

2.2.2 *Confronto di alcune blockchain*

Di seguito una comparazione di tre grandi blockchain moderne con UniBlock:

	Bitcoin	***Ethereum***	***Hyperledger Fabric***	***UniBlock***
Accesso	Pubblica	Pubblica	Privata	Ibrida
Consenso	Proof of work	Proof of work	Crash Fault Tolerance	Proof of work
Privacy	Non crittografata	Non crittografata	Non crittografata	Crittografata

2.3 CRITTOGRAFIA

2.3.1 *Hash crittografico*

Una funzione hash è una funzione che trasforma un input di dimensione arbitraria in un output di dimensione fissa. Per poter essere impiegata in crittografia una funzione hash deve rispettare forti vincoli di sicurezza:

RESISTENZA ALLA PRE-IMMAGINE Dato un particolare hash h, dev'essere estremamente difficoltoso trovare un input che abbia come hash h. La funzione di hash deve quindi risultare una funzione one-way, cioè una funzione la cui inversa è impossibile da costruire o computazionalmente non eseguibile in tempi accettabili.

SECONDA RESISTENZA ALLA PRE-IMMAGINE Dato un un particolare input a, dev'essere estremamente difficoltoso trovare un altro input che abbia lo stesso hash di a.

RESISTENZA ALLE COLLISIONI Dev'essere estremamente difficoltoso trovare due input che abbiano lo stesso hash.

2.3.2 *Crittografia simmetrica*

Gli algoritmi a chiave simmetrica impiegano la stessa chiave per cifrare un testo e per decifrarlo successivamente. Le due tipologie principali di algoritmi a chiave simmetrica sono:

CIFRARI A BLOCCHI L'algoritmo prende in input un numero fisso di bit e li cripta combinandoli con la chiave restituendo lo stesso numero di bit criptati in uscita. Gli unici input accettabili sono di dimensione pari al blocco di bit usato, qualunque altro input dovrà essere opportunamente allungato se troppo corto o troncato se troppo lungo. Solo usando l'algoritmo in particolari modalià di operazione potrà essere possibile permettere input di dimensione arbitraria.

CIFRARI A FLUSSO L'algoritmo combina l'input con un flusso pseudo-random di bit derivato dalla chiave. Al contrario dei cifrari a blocchi non presentano problemi di dimensione dell'input, in quanto il flusso generato verrà calibrato esattamente della lunghezza necessaria per ciascun input.

2.3.3 *Modalità di operazione*

Data l'estrema restrittiva dei cifrari a blocchi che permettono di criptare solo un blocco di bit di dimensione fissa, sono stati concepiti diversi modi di usare tali cifrari per garantirne certe proprietà o per renderne più agevole l'uso in alcune situazioni:

ECB L'input viene partizionato in blocchi di dimensione fissa e ognuno viene criptato singolarmente.

CBC L'input viene partizionato in blocchi di dimensione fissa e criptato mettendolo in xor con il blocco criptato precedente.

CTR L'input viene partizionato in blocchi di dimensione fissa e ogni blocco viene criptato con una chiave generata a partire da una funzione sequenziale. Permette di fatto di trasformare un cifrario a blocchi in uno a flusso.

GCM Modalità molto simile a CTR in quanto permette tramite la derivazione sequenziale di una chiave di trasformare un cifrario a blocchi in uno a flussi. Aggiunge mediante l'uso dei campi di Galois una firma che garantisce l'integrità del processo di cifrazione.

2.3.4 *Crittografia asimmetrica*

Gli algoritmi a chiave pubblica impiegano due chiavi diverse, una per cifrare e una per decifrare. Quella per cifrare è chiamata chiave pubblica

in quanto non costituisce segreto ed è liberamente distribuibile senza protezioni. Quella per decifrare invece è chiamata chiave segreta e deve essere mantenuta appunto segreta e nota solo alla persona che la possiede. Tra le due chiavi è presente una relazione matematica che solitamente implica una funzione one-way sottostante, in quanto deve essere molto facile generare una chiave pubblica a partire da una chiave privata ma il contrario non deve essere computazionalmente fattibile. Oltre che cifrare è possibile anche firmare un input con la propria chiave privata così da garantire la sua autenticità. Chiunque per mezzo della chiave pubblica potrà verificare infatti che la firma proviene dalla chiave privata corrispondente.

2.3.5 *Crittografia ellittica*

La crittografia ellittica è una particolare famiglia di algoritmi a chiave pubblica che si basano sull'uso di curve ellittiche $y^2 = x^3 + ax + b$ su campi finiti. La funzione one-way che protegge la relazione tra le due chiavi è quella del logaritmo discreto. Al contrario di altri algoritmi a chiave pubblica come RSA (basato sulla fattorizzazione in numeri primi) permettono di avere chiavi di dimensioni molto più contenute e un'efficienza di esecuzione molto elevata.

2.3.6 *Scambio di chiavi tramite Diffie-Hellman*

Lo scambio di chiavi tramite Diffie-Hellman permette a due soggetti di scambiarsi tramite un canale insicuro le proprie chiavi pubbliche, combinandole con le chiavi private in modo tale da giungere ognuno separatamente a una certa quantità. Tale quantità risolterà identica per entrambi grazie alle proprietà matematice che collegano le chiavi in gioco. Un possibile attaccante in ascolto sul canale non potrà calcolare la stessa quantità in quanto le rispettive chiavi private non hanno mai transitato nel canale, risultando quindi ignote all'attaccante e impossibili da ottenere a partire da quelle pubbliche. La quantità comune ottenuta è tipicamente utilizzata come base per derivare una chiave simmetrica.

2.3.7 *Algoritmi impiegati*

SHA-3 Algoritmo di hashing crittografico pensato come futuro successore della famiglia di algoritmi SHA-2 [4]. Usato in UniBlock come

puzzle crittografico alla base della proof of work.

AES Algoritmo di cifratura simmetrica a blocchi diventato lo standard de facto per questo genere di algoritmi avendo supporto a livello di assembly nella maggior parte delle CPU moderne [8]. La dimensione dei blocchi è pari a 128 bit. Usato in UniBlock in modalità GCM per criptare il contenuto degli eventi.

X25519 Algoritmo di scambio di una chiave tramite Diffie-Hellman basato sulla curva ellittica Curve25519 [1]. Utilizzato in UniBlock per negoziare tra due utenti una chiave comune.

ED25519 Algoritmo di firma a chiave pubblica basato sulla curva ellittica Curve25519 [2]. Usato in UniBlock per firmare gli eventi generati da un utente.

BIBLIOGRAFIA

[1] Daniel J. Bernstein. Curve25519: New diffie-hellman speed records. In *Public Key Cryptography - PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 207–228. Springer, 2006. doi: 10.1007/11745853_14. URL https://iacr.org/archive/pkc2006/39580209/39580209.pdf. (Cited on page 11.)

[2] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2):77–89, Sep 2012. ISSN 2190-8516. doi: 10.1007/s13389-012-0027-1. URL https://doi.org/10.1007/s13389-012-0027-1. (Cited on page 11.)

[3] Natalia Chaudhry and Muhammad Murtaza Yousaf. Consensus algorithms in blockchain: Comparative analysis, challenges and opportunities. In *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*, pages 54–63, Dec 2018. doi: 10.1109/ICOSST.2018.8632190. (Cited on page 7.)

[4] Morris Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions, 2015-08-04 2015. (Cited on page 10.)

[5] EthHub. Privacy on ethereum. https://docs.ethhub.io/ethereum-roadmap/privacy/. [Online; accessed 08-06-2022]. (Cited on page 7.)

[6] Christine V. Helliar, Louise Crawford, Laura Rocca, Claudio Teodori, and Monica Veneziani. Permissionless and permissioned blockchain diffusion. *International Journal of Information Management*, 54:102136, 2020. ISSN 0268-4012. doi: https://doi.org/10.1016/j.ijinfomgt.2020.102136. URL https://www.sciencedirect.com/science/article/pii/S0268401219314586. (Cited on page 6.)

[7] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, 2008. URL https://bitcoin.org/bitcoin.pdf. (Cited on page 5.)

[8] Information Technology Laboratory (National Institute of Standards and Technology). Announcing the advanced encryption standard (aes). Technical report, 2001. URL https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf. (Cited on page 11.)