CryptoNight & RandomX

Alessia Angelone, Francesco Baccaro, Simona Bertè,
Emilio Cassaro, Christian Coduri, Giovanni Nicosia

13 Giugno 2024

Indice

Capitolo 1

Introduzione

1.1 Protocollo di consenso

In una rete blockchain pubblica, nessuna singola entità detiene esclusivamente l'autorità di verificare e validare le transazioni, in quanto un avversario potrebbe falsificare una catena di blocchi con transazioni false e propagare la blockchain fasulla nella rete. Pertanto, un componente fondamentale delle reti blockchain è il protocollo di consenso, che consiste in un insieme di regole e procedure utilizzate per raggiungere un accordo comune, tra tutti i nodi partecipanti alla rete, sullo stato della blockchain.

1.1.1 Proof Of Work (PoW)

Uno dei protocolli di consenso più utilizzati è il Proof-of-Work (PoW) che si basa sul dimostrare di aver svolto uno sforzo computazionale per creare un blocco che soddisfa il protocollo di consenso. Questo processo, che porta ad avere un nuovo blocco valido da inserire nella blockchain, prende il nome di *mining*. In base alla blockchain considerata, il minatore che ha inserito il blocco può raccogliere le ricompense predefinite e/o le commissioni delle transazione contenute in tale blocco.

L'idea iniziale era che ogni partecipante potesse unirsi al processo di mining utilizzando le proprie risorse computazionali. In questo modo, un avversario per violare il protocollo PoW, deve necessariamente possedere una potenza di calcolo superiore a quella aggregata di tutti gli altri partecipanti al mining (attacco del 51%). Tuttavia, questa minaccia inizia a diventare sempre più reale a causa della diffusione di miner basati su circuiti integrati specifici.

1.2 Mining: CPU vs GPU vs FPGA vs ASIC

Tra le tecnologie maggiormente usate per il mining possiamo distinguere: ASIC (Application-Specific Integrated Circuits), FPGA (Field-Programmable Gate Arrays), GPU (Graphics Processing Units), e CPU (Central Processing Units).

Ognuna di queste tecnologie viene impiegata per scopi precisi e presenta delle limitazioni rispetto alle altre. Pertanto, dato che scegliere l'una piuttosto che l'altra comporta differenze in termini di efficienza e costi, è importante avere un idea generale delle loro differenze.

- Le **CPU**, le "generaliste del gruppo" presenti in ogni personal computer, offrono grande flessibilità, in quanto sono in grado di gestire un'ampia gamma di compiti a scapito dell'efficienza specializzata riscontrabile nelle altre tecnologie.

- Le **GPU**, inizialmente sviluppate per il rendering grafico, hanno trovato applicazioni diverse grazie alla loro capacità di elaborazione parallela, dimostrandosi strumenti potenti nei calcoli ad alta intensità di dati.

Nel contesto del mining di criptovalute, CPU e GPU sono state le prime scelte, ma con l'aumento della "concorrenza", ASIC e FPGA come soluzioni più efficienti.

- Gli **FPGA (*Field Programmable Gate Arrays*)** possono implementare una vasta gamma di funzioni circuitali, in quanto sono fondamentalmente dispositivi modificabili fisicamente per creare lo specifico circuito elettronico che si desidera.

Poiché devono essere versatili, i componenti presenti internamente hanno molte caratteristiche che non vengono sfruttate. Di conseguenza, nonostante possano raggiungere prestazioni superiori ad un computer general purpose, non sono ottimizzati per il consumo energetico e la velocità in un'applicazione specifica come quella del mining.

- Un **ASIC (*Application-specific integrated circuit*)** è simile ad un FPGA in quanto utilizza un circuito elettronico cablato per eseguire una funzione computazionale. A differenza di un FPGA, un ASIC può essere ottimizzato per il consumo energetico ridotto, la velocità o una combinazione di entrambi, comportandosi quindi sempre meglio di un FPGA programmato per fare lo stesso lavoro.

  Gli svantaggi di un ASIC sono il costo di ingresso elevato ed il complesso processo di progettazione necessario per il loro sviluppo.

  La non modificabilità degli ASIC è il loro punto di forza in termini di consumo energetico e velocità, ma è anche una loro criticità. Essendo progettati per eseguire un compito specifico, non possono essere adattati o aggiornati per eseguire funzioni diverse da quelle per cui sono stati progettati. Pertanto, se una blockchain dovesse cambiare l'algoritmo di mining, gli ASIC progettati per minare su essa, diventerebbero obsoleti ed inefficaci poiché non modificabili.

Nonostante quest'ultima problematica, gli ASIC restano la minaccia più grande alle blockchain, in quanto offrono prestazioni superiori e un consumo energetico ottimizzato specificamente per il mining. Con il predominio di questi circuiti integrati, appositamente progettati per il calcolo di specifici PoW, la decentralizzazione delle reti blockchain e la looro integrità è messa a rischio.

Caso Bitcoin

Osservando il grafico in Figura 1.1, si può vedere come l'hashrate aggregato della rete Bitcoin, ovvero il numero di hash che vengono prodotti globalmente al secondo, è aumentato drammaticamente.

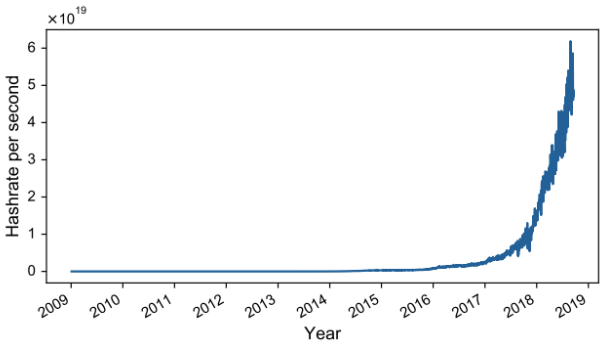

Figura 1.1: Hashrate della blockchain Bitcoin [1]

La motivazione riconducibile a ciò è proprio l'avvento degli ASIC e la loro partecipazione al mining di Bitcoin.

Il mining basato su ASIC ha creato un picco barriera all'ingresso per il grande pubblico perché, chi interessato a minare, dovrebbe investire in attrezzature speciali per partecipare a tale processo. A causa di ciò, come precedentemente accennato, le poche entità in grado di investire e mantenere un grande volume di ASIC potrebbero assumere il controllo della rete. Pertanto, il mining basato su ASIC è maggiore vulnerabile all'attacco del 51%.

A partire da settembre 2018, i mining-pool[1] Bitcoin BTC.com e Antpool, che sono gestiti dalla stessa azienda che produce gli ASIC per la blockchain bitcoin, rappresentano oltre il 30% del potere (hashrate) nella rete Bitcoin.

[1]**Mining-pool**: minatori di criptovalute che uniscono le loro risorse per estrarre criptovalute e aumentare le probabilità di ottenere la ricompensa di un blocco

Capitolo 2

CryptoNote

Capitolo 3

CryptoNight

3.1 Introduzione alle PoW ASIC-resistant

Al fine di scoraggiare l'uso di sistemi basati su ASIC per il mining, citati nel primo capitolo, sono stati proposti diversi meccanismi PoW ASIC-resistant. I principali algoritmi di questo tipo possono essere divisi in: PoW Multi-hash, PoW Memory-hard e PoW Programmatico.

3.1.1 PoW Multi-hash

A differenza dell'algoritmo PoW di Bitcoin che utilizza solo un tipo di funzione hash, gli algoritmi PoW Multi-hash impiegano più funzioni hash per calcolare la validità del blocco. Queste funzioni hash vengono applicate all'intestazione del blocco in una sequenza, che può essere fissa o determinata dinamicamente per ogni blocco. Alcuni degli algoritmi PoW multi-hash più famosi sono: *X11, X14, X17, X11EVO, X16S, X16R, Quark* e *TimeTravel*.

Nella ricerca scientifica di Cho, dal titolo "*ASIC-Resistance of Multi-Hash Proof-of-Work Mechanisms for Blockchain Consensus Protocols*" [1], è stato dimostrato come gli algoritmi multi-hash non hanno una significativa differenza da altri meccanismi PoW semplici. Inoltre, nonostante gli algoritmi di questa categoria siano spesso definiti ASIC-resistance, alcuni di loro sono già stati rotti da sistemi ASIC. Mentre, per quanto riguarda gli algoritmi rimanenti, non è stato mai dimostrato se siano veramente resistenti agli ASIC o se semplicemente non sono ancora stati progettati ASIC specifici per essi, magari in quanto impiegati in reti piccole dal basso valore economico.

Dati i risultati ottenuti, nel paper si suggerisce di optare per l'impiego di altri meccanismi ASIC-resistant per scoraggiare effettivamente il mining basato su ASIC.

3.1.2 PoW Memory-hard

Sebbene gli ASIC offrano una maggiore efficienza computazionale rispetto alle piattaforme di calcolo general-purpose, sono solitamente limitati in termini di memoria.

In un algoritmo PoW Memory-hard, il processo di mining richiede l'utilizzo di grandi quantità di memoria per importare dati complessi e calcolarne l'hash. Una computazione in questi algoritmi richiederà il recupero di dati casuali da un grande set di dati. La dimensione dell'insieme sarà sufficientemente grande da rendere difficile la progettazione di un ASIC che memorizzi tutto in una memoria on-chip.

Gli algoritmi PoW memory-hard più conosciuti sono: *Ethash* di Ethereum, *Scrypt* e *CryptNight*.

3.1.3 PoW Programmatico

Infine, una delle nuove direzioni per meccanismi PoW resistenti è aumentare la diversità delle computazioni. Nell'approccio PoW Programmatico, il processo di mining coinvolge l'esecuzione di un programma che cambia frequentemente. Ad esempio, aggiungendo come parte della computazione un pool di funzioni matematiche o un programma generato casualmente.

Questo rende difficile per gli ASIC ottimizzare il mining, poiché dovrebbero essere riprogrammati ogni volta che il programma cambia. Sarebbe impraticabile costruire moduli hardware specializzati che mirino a ciascuna delle possibili attività di calcolo.

Questo tipo di meccanismi PoW sono in fase di studio e non sono ancora stati utilizzati per la realizzazione di una vera e propria blockchain.

3.2 CryptoNight Hash Function

Come descritto precedentemente per gli algoritmi PoW memory-hard, i processori specializzati quali ASIC e FPGA hanno un vantaggio minimo rispetto alle CPU e GPU a causa della complessità degli algoritmi e dell'elevato costo della memoria.

Ad esempio, nella blockchain Ethereum l'algortimo di consenso è *Ethash*. L'Antminer E3 sviluppato per minare su Ethereum ha un hash rate di 190 MH/s, che è solo 2 volte più veloce della GPU V100, a fronte di un consumo energetico 3 volte superiore.

3.2.1 Algoritmo CryptoNight

Un altro algoritmo di PoW molto conosciuto, classificabile come memory-hard, è CryptoNight. Tale algoritmo è compreso nel protocollo di consenso CryptoNote rilasciato nel 2013 e cerca di offrire un'**elevata dipendenza da CPU**, resistendo ad ASIC, FPGA e GPU.

Si ricordi che le CPU moderne possono essere costituite da un massimo di qualche decina di core e possono utilizzare più livelli di cache per migliorare la velocità di accesso alla memoria. Nello specifico, le cache L1 e L2 sono private per ogni core, mentre la cache L3 è condivisa da tutti i core.

CryptoNight contiene un ciclo memory-hard, che esegue una sequenza di letture e scritture casuali in una piccola area di memoria da 2MB, chiamata *scratchpad*. Questo ciclo è stato progettato per funzionare in modo efficiente sulle CPU perché lo scratchpad è dimensionato per adattarsi alla **cache L2** della CPU.

L'algoritmo CryptoNight si compone di tre fasi principali: inizializzazione dello scratchpad, il memory-hard loop e calcolo del risultato.

Fase 1: Inizializzazione dello scratchpad

Utilizzando la funzione Keccak (SHA3) vengono generati 200 byte.

- I primi 32 byte (0-31) vengono utilizzati come chiave AES-256.
- I successivi 128 byte (64-191), vengono divisi in otto blocchi di 16 byte ciascuno, e ciascuno di questi otto blocchi viene cifrato per 10 round AES utilizzando la chiave AES-256.
- Il risultato della cifratura viene memorizzato come i primi 128 byte della memoria dello scratchpad. Successivamente, viene eseguita iterativamente la cifratura AES sui 128 byte correnti dello scratchpad e il risultato viene aggiunto come i successivi 128 byte nello scratchpad.
- Questo processo continua fino a quando l'intero scratchpad è riempito.

Fase 2: Memory-hard loop

Come elencato nell'Algoritmo XX e illustrato nella Figura 3.1.

- Sia A che B sono interi a 16 byte e vengono inizializzati con il risultato XOR dei byte da 0 a 31 e da 32 a 63 generati utilizzando la funzione Keccak.
- Questi interi vengono quindi utilizzati come indirizzi nello scratchpad analizzando i 21 bit meno significativi. Si legge il valore S[A] nello scratchpad usando A come indirizzo e si esegue la crittografia AES di S[A] con A per ottenere il risultato C.
- Successivamente si esegue lo XOR su B e C e si riscrive il risultato XOR in S[A]. Si usa poi C come indirizzo, si legge S[C] in D, si moltiplicano C e D, si aggiunge il risultato della moltiplicazione ad A e si memorizza A in S[C].
- Infine, si ottiene il risultato XOR di A e D come nuovo A, e C come nuovo B, e si utilizeranno il nuovo A e B all'iterazione successiva.

Poiché l'output della crittografia AES è casuale, non è possibile prevedere l'indirizzo, quindi il ciclo non può essere parallelizzato. Il ciclo contiene 524.288 iterazioni, quindi verrà eseguita una sequenza di 2 milioni di letture e scritture casuali complessive sullo scratchpad.

Algorithm 1

Require: Integer A, B, Scratchpad memory S
Ensure: Modified scratchpad memory S
1: **for** $i \leftarrow 1$ to 524288 **do**
2: $C \leftarrow \text{AES}(S[A], A)$
3: $S[A] \leftarrow \text{XOR}(B, C)$
4: $D \leftarrow S[C]$
5: $A \leftarrow \text{ADD}(A, \text{MUL}(C, D))$
6: $S[C] \leftarrow A$
7: $A \leftarrow \text{XOR}(A, D)$
8: $B \leftarrow C$
9: **end for**

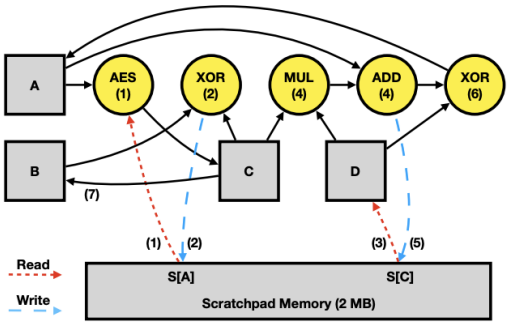

Figura 3.1: Memory-Hard loop dell'algoritmo CryptoNight. [2]

Fase 3: Calcolo del risultato

Similmente al primo passo, i byte da 32 a 63 del risultato iniziale di Keccak vengono utilizzati come chiave AES-256 e i byte da 64 a 191 vengono crittografati con il contenuto della memoria scratchpad, 128 byte alla volta.

Successivamente, la permutazione Keccak viene eseguita una volta sul risultato della crittografia. A seconda dei due bit meno significativi del primo byte del risultato Keccak, verrà scelto uno dei quattro algoritmi di hash: BLAKE-256, Groestl-256, JH-256 o Skein256. L'algoritmo di hash selezionato viene applicato sul risultato Keccak per produrre l'output finale di CryptoNight.

Con CryptoNight come protocollo di consenso della blockchain, agli utenti che richiedono di aggiungere un nuovo blocco verrà richiesto di eseguire ripetutamente l'algoritmo CryptoNight fino a trovare un nonce n tale che $H(n||b) \times d < 2^{256}$, dove H è la funzione di hash CryptoNight, b è il contenuto del nuovo blocco e d è la difficoltà.

Quando un nuovo blocco viene trasmesso alla rete blockchain, il validatore eseguirà la funzione di hash CryptoNight sul nuovo blocco per verificare se il valore di hash del nuovo blocco sia effettivamente inferiore alla soglia data.

L'algoritmo Cryptonight non è considerato un algoritmo multi-hash nel senso tradizionale. Anche se potrebbe utilizzare più passaggi di hash all'interno del suo processo di computazione, la sua caratteristica distintiva è la sua natura memory-hard .

Risultati sperimentali

Nel paper intitolato "*Evaluating Memory-Hard Proof-of-Work Algorithms on Three Processors*" [2], Feng e Luo confrontano tra loro *CryptoNight*, *Ethash*, *Hashcash* e *Cuckoo* su CPU, GPU e KNL, giungendo alla conclusione che la Memory-Hardness può essere raggiunta sfruttando la latenza o la bandwith.

I compiti che dipendono dalla latenza sono più adatti per CPU, grazie alla loro gerarchia di cache ben sviluppata. Al contrario, i compiti che richiedono una grande larghezza di banda possono sfruttare appieno il potenziale delle GPU.

Nel caso di studio, CryptoNight implementa una resistenza alla memoria mediante la latenza, sfruttando il tempo di accesso alla memoria per rendere più difficile la creazione di hardware specializzato.

Tuttavia, come seconda considerazione, viene specificato che, poiché i processori si stanno evolvendo rapidamente, gli algoritmi di Proof-of-Work dovrebbero essere adattabili e flessibili affinché le loro proprietà possano essere mantenute il più a lungo possibile.

3.2.2 La vittoria degli ASIC su CryptoNight

CryptoNight rimase immune agli ASIC per un lungo periodo. Tuttavia, a partire dal 2018 vennero annunciati diversi modelli di ASIC per CryptoNight (Bitmain, Baikal e Halong Mining), capaci di raggiungere un hashrate superiore ai 200 KH/s.

Bitmain

Il 15 marzo 2018, Bitmain, un'azienda cinese, ha annunciato l'Antminer-X3, un ASIC progettato appositamente per il mining di criptovalute basate su CryptoNight. Secondo le specifiche riportate sul sito di Bitmain, l'X3 ha un tasso di hash totale di 220 kH/s e un consumo energetico di 465W.

È interessante notare la risposta su Twitter del responsabile del progetto Monero all'epoca, Riccardo Spagni, il quale sottolinea l'inefficacia dell'Antminer-X3 per la blockchain di Monero (Figura 3.2). Al tempo Monero utilizzava come PoW proprio CryptoNight, tuttavia come parte del suo protocollo blockchain adotta regolarmente hard fork pianificati e di emergenza al fine di mitigare qualsiasi potenziale minaccia derivante dagli ASIC.

Figura 3.2: Modello Antminer-X3 (Bitmain)

La strategia adottata da Monero gli conferisce immunità agli ASIC come l'X3 e quelli che presentati nelle prossime sezioni. Tuttavia, tali dispositivi rimangono utili per il mining di altre criptovalute basate su CryptoNight, come Bytecoin, Dinastycoin, Karbo e vecchie fork di Monero, che non implementano misure simili.

Baikal

Anche la società russa Baikal Mining ha reagito prontamente, rilasciando un primo modello meno potente nello stesso mese del concorrente cinese e un secondo modello più performante alcuni mesi dopo. Entrambi gli ASIC sono in grado di minare gli algoritmi CryptoNight e CryptoNight-Lite.

(a) Modello BK-N, rilasciato nel marzo 2018 con un hashrate massimo di 80 kH/s ed un consumo di 120W (Figura 3.3a),

(b) Modello BK-N240, rilasciato nel maggio 2018 con un hashrate massimo di 480 kH/s ed un consumo di 650W (Figura 3.3b).

(a) (b)

Figura 3.3: Modelli ASICS BK-N e BK-N240 (Baikal)

Halong Mining

Con un mese di ritardo, nell'Aprile 2018, anche l'azienda statunitense Halong Mining ha rilasciato il suo Modello DragonMint X2, in grado di minare l'algoritmo CryptoNight con un hashrate massimo di 248 kH/s e un consumo energetico di 490W.

Figura 3.4: Modello ASICS DragonMint X2 (Halong Mining)

Capitolo 4

Privacy

Preservare la privacy degli utenti è fondamentale nelle reti informatiche e diventa ancor più cruciale nei sistemi pubblici, aperti e decentralizzati, come le blockchain.

Tre sono i principali rischi per la privacy:

1. **Locali**, dovuti all'uso di PC/Smartphone (browser history, cookies, blockchain client).
2. **Network** (informazioni rivelate dall'indirizzo IP, man-in-the-middle, condivisione inavvertita dei dati privati)
3. **Blockchain** (riutilizzo degli stessi address, analisi a ritroso delle transazioni).

Quest'ultima può essere facilmente compromessa attraverso il clustering degli indirizzi. Questa tecnica consiste nel trovare e raggruppare gli indirizzi in wallet, ossia gruppi di indirizzi presumibilmente appartenenti a un singolo soggetto. Il clustering si realizza tramite l'analisi della blockchain e l'utilizzo di approcci euristici, descritti da Jonas David Nick nel suo lavoro "Data-Driven De-Anonymization in Bitcoin".

4.1 Euristiche

Le euristiche sono misurazioni basate sull'analisi del protocollo e sull'esperienza. Non sono sempre accurate, ma se ben calibrate, possono fornire un'indicazione del livello di attendibilità dei risultati ottenuti.

La prima euristica è chiamata "Multi-Input Heuristic". In accordo con quanto affermato da Satoshi Nakamoto, inventore di Bitcoin, questa regola mostra come tutti gli indirizzi in input di una transazione provengano dallo stesso wallet, poiché l'autore della transazione possiede le chiavi private di tutti gli indirizzi in input. Tuttavia, l'autore non è necessariamente una singola persona; potrebbe trattarsi di un exchange, un mixer o un wallet online.

La seconda euristica, denominata "Shadow Heuristic", si basa su come i wallet gestiscono le transazioni con resto. E' bene osservare che nella maggior parte dei wallet, quando un utente invia una transazione, viene generata una nuova coppia di chiavi (pubblica e privata) per gestire l'output di resto. In una tipica transazione con un input e due output, uno degli output rappresenta l'indirizzo di destinazione e l'altro è l'indirizzo di resto. Supponendo che l'indirizzo di destinazione appartenga a un commerciante o a un'entità che riceve frequentemente pagamenti, questo indirizzo rimarrà costante in diverse transazioni. L'altro output, che cambia ad ogni transazione, è l'indirizzo di resto generato dal wallet. Da qui si evince che le chiavi pubbliche di invio e di resto siano nello stesso wallet. Di conseguenza, un potenziale indirizzo di resto può essere identificato come un indirizzo che appare in massimo due transazioni: una come output (quando riceve il resto) e una come input (quando il resto viene speso).

La terza euristica, la "Consumer Heuristic", è un'altra tecnica utilizzata nell'analisi della blockchain per identificare le transazioni di resto e collegare indirizzi appartenenti allo stesso wallet, basandosi sul comportamento tipico degli utenti privati nel gestire le loro transazioni. Gli utenti privati tendono a effettuare transazioni con un solo destinatario alla volta. Pertanto, le transazioni effettuate da questi utenti hanno tipicamente al massimo due output: uno per il destinatario e uno per il resto. La sfida è identificare quale dei due output è il resto, analizzando le transazioni successive. Se in una transazione successiva, uno degli output dubbi invia denaro a più destinatari, è probabile che questo output sia il resto. Questo perché il resto, una volta tornato nel wallet dell'utente, può essere utilizzato in future transazioni che potrebbero avere più di un output. L'altro

output, che non viene utilizzato in transazioni con più destinatari, è più probabilmente quello del destinatario originale della transazione iniziale.

L'ultima euristica è la "Optimal Change Heuristic", basata sul fatto che i wallet cercano di ottimizzare la gestione del resto per ridurre le dimensioni delle transazioni e le conseguenti commissioni, minimizzando il numero di input e output necessari per ogni transazione. Da questo segue una regola generale in cui, come spiegato da Jonas Nick, se c'è un unico output con un valore inferiore a qualsiasi input, allora quello sarà il resto della transazione.

Questi approcci, insieme a principi più complessi, sono impiegati dai principali strumenti open source, come BitCluster e WalletExplorer, per supportare strategie efficaci di clustering. Combinando questi strumenti con un monitoraggio costante delle transazioni su piattaforme online per la visualizzazione della blockchain, o scaricando una copia della blockchain e installando un block explorer locale, è possibile ottenere ottimi risultati. Questi metodi permettono di collegare indirizzi ignoti a indirizzi noti, ricostruendo indirettamente l'identità dei proprietari o almeno i movimenti di denaro.

4.2 Soluzioni

Esistono diverse soluzioni per affrontare il problema della privacy nelle transazioni blockchain. Tra di esse ricordiamo la Ring Confidential Transaction, che permette a chi paga di dimostrare di avere a disposizione il denaro, ma senza indicare in chiaro l'importo della transazione; la Stealth Address, ovvero l'utilizzo di indirizzi indivisibili al fine di rendere difficile risalire ai wallet coinvolti nella transazione; Kovri, che consente agli utenti di nascondere il proprio indirizzo IP; lo sviluppo stesso di wallet e protocolli che implementano politiche di gestione del resto più complesse, al fine di rendere le transazioni meno suscettibili all'analisi del clustering. La soluzione su cui ci vogliamo soffermare riguarda l'utilizzo di tecniche avanzate di anonimizzazione, come le **firme ad anello**, i protocolli di **mixing** o **CoinJoin**, che offuscano le relazioni tra input e output delle transazioni, rendendo più difficile collegare gli indirizzi di un singolo utente.

Quando si parla di **mixer** o **tumbler**, si fa riferimento a servizi gestiti da terzi che "mescolano" le criptomonete inviate al fine di renderle anonime e difficili da tracciare. Questi servizi "ripuliscono" il denaro facendo in modo che passi attraverso vari indirizzi non collegati tra loro, se non tramite gli algoritmi interni del servizio stesso. Il funzionamento dei servizi di mixing è piuttosto semplice: l'utente invia l'importo che desidera "ripulire" all'indirizzo Bitcoin fornito dal mixer. Il mixer, dopo aver trattenuto una piccola commissione, invia l'importo all'indirizzo indicato dall'utente, ma questi fondi provengono dai versamenti di altri utenti. In questo modo, diventa molto difficile, se non impossibile, collegare i due indirizzi tramite un'analisi della blockchain. I migliori servizi di mixing offrono ulteriori funzionalità per aumentare l'anonimato, come il ritardo temporale e la scomposizione delle transazioni. Ad esempio, alcuni servizi, come BitMixer, permettono di dividere la transazione in più parti, ognuna delle quali viene inviata all'indirizzo di destinazione con un ritardo temporale stabilito dall'utente. Questo significa che può trascorrere anche qualche giorno tra la ricezione della prima parte dei fondi e il completamento della transazione. Naturalmente, il servizio ha un costo, sebbene limitato: sulla transazione viene applicata una piccola commissione, fissa o variabile, che viene ulteriormente anonimizzata attraverso vari metodi. La sicurezza e l'anonimità di questi servizi centralizzati è ovviamente discutibile. Gli utenti non hanno alcuna garanzia che il mixer restituisca il loro denaro o che le monete rese non siano state segnate in qualche modo. Un altro aspetto da considerare quando si usa un mixer è che l'indirizzo IP e l'indirizzo Bitcoin potrebbero essere registrati da una terza parte.

Un'altra possibile soluzione è l'utilizzo di operazioni di **CoinJoin**, che permettono di superare le criticità e i rischi associati ai mixer. CoinJoin consente di combinare la propria transazione con quella di altri utenti, senza dover riporre fiducia in un ente terzo, potenzialmente malevolo. Il principio di CoinJoin è che più utenti si coordinino per creare una singola transazione, ciascuno fornendo i propri input e output desiderati. Poiché tutti gli input vengono mescolati, diventa impossibile determinare con certezza quale output appartiene a quale utente. Al massimo, si può dire che un partecipante ha fornito uno degli input e potrebbe essere il proprietario di uno degli output, ma anche questo non è garantito. Un coordinatore raccoglie tutte le informazioni necessarie, crea la transazione e la fa firmare a ciascun partecipante prima di trasmetterla alla rete. Una volta firmata, la transazione non può essere modificata senza diventare invalida, eliminando così il rischio che il coordinatore possa fuggire con i fondi. I vantaggi di CoinJoin sono principalmente due: l'incremento della privacy e il costo contenuto. CoinJoin offre un elevato livello di privacy combinando le transazioni di più utenti, rendendo difficile tracciare l'origine e la destinazione dei fondi. Inoltre, non introduce costi aggiuntivi significativi rispetto a una transazione normale,

il che lo rende economicamente conveniente. Tuttavia, CoinJoin presenta alcune limitazioni, in particolare riguardo la scalabilità. È necessario che più utenti si coordinino e sincronizzino le proprie transazioni per partecipare efficacemente a CoinJoin. Questo processo di coordinazione può essere complesso e meno pratico su larga scala, rendendo CoinJoin meno adatto per situazioni in cui è richiesta una grande partecipazione.

Una terza possibile soluzione consiste nell'utilizzo delle **firme ad anello** (Riferimento spiegazione ring signature), caratteristica fondamentale di molte criptomonete basate sul protocollo CryptoNote.

L'utilizzo di firme ad anello è vantaggioso rispetto alle altre soluzioni: Coinjoin e mixing si basano su terze parti, mentre CryptoNote permette, una volta posseduta la blockchain, di eseguire il mixing localmente. In generale le firme ad anello forniscono un livello di privacy superiore poiché il mixing è integrato nel protocollo. Dall'altra parte, sebbene un anello di dimensioni maggiori migliori l'irrintracciabilità, aumenta anche il costo di convalida della transazione, poiché la commissione è proporzionale alla dimensione della transazione, che cresce con la dimensione dell'anello.

4.3 Scelta dei mixin nel protocollo Cryptonote

Supponiamo che Alice voglia trasferire criptomonete a Bob. Essa dovrà utilizzare i suoi output di transazione (TXO) e le rispettive chiavi private per creare una nuova transazione. Per rendere la transazione irrintracciabile, Alice dovrà selezionare $l-1$ TXO di depistaggio, detti decoy, dalla blockchain per ogni TXO reale che vuole trasferire. Dovrà poi associare ogni TXO reale a $l-1$ depistaggi, creando una firma ad anello con il gruppo $G_i = \{g_1, g_2, \ldots, g_l\}$, i cui elementi sono chiamati **mixin**. Grazie alla firma ad anello one-time, Alice potrà firmare la transazione con le sue chiavi private, dichiarando i mixin come potenziali spese. In particolare, poiché tutti i TXO in input di una transazione devono avere la stessa denominazione, il software client manterrà un database di decoy disponibili, indicizzati per denominazione, da cui campionare i mixin . Alice potrà regolare individualmente le sue politiche di selezione, in quanto la transazione verrà accettata dai miner solo se gli elementi di G_i esistono sulla blockchain ed essa è in possesso della chiave privata per almeno uno di essi, indipendentemente dalla distribuzione utilizzata nella selezione dei decoy. Seppur l'anonimizzazione è mantenuta dal fatto che i miner non passono identificare quale elemento di G_i è stato effettivamente speso in quella transazione, mantenendo anonimi gli input all'interno del gruppo di l input potenziali, è cruciale che l'utente scelga attentamente i mixin.

Il protocollo Cryptonote non fornisce una raccomandazione esplicita su come i "mixin" dovrebbero essere scelti. L'implementazione di riferimento originale di Cryptonote includeva una politica di selezione "uniforme", che è stata adottata (almeno inizialmente) dalla maggior parte delle implementazioni, incluso Monero. Solo successivamente quest'ultimo ha cambiato algoritmo, adottando la distribuzione Gamma nella selezione dei decoy, come vedremo in seguito.

4.4 Attacks

Prima di procedere, diamo la seguente definizione.

Definition 1 *Si definisce **età di un mixin** la differenza tra l'altezza del blocco in cui viene utilizzato come mixin e l'altezza del blocco in cui viene prodotto, ovvero minato come output di una transazione.*

In altre parole, se un TXO è registrato come la N-esima transazione sulla blockchain e appare nel set di mixin della M-esima transazione sulla blockchain, allora definiamo l'età di quel mixin come $M - N$.

Nel corso degli ultimi anni, differenti studi [**?**]-[**?**] hanno mostrato che il modo in cui gli utenti selezionano i decoy delle loro transazioni può influenzare la loro privacy e quella degli altri.

Gli autori del [**?**], in cui è possibile trovare maggiori dettagli, hanno dimostrato che l'età di un TXO, al momento del trasferimento da parte del proprietario, segue una distribuzione simile alla distribuzione Gamma. Pertanto, se l'età dei decoy non segue la stessa distribuzione, una semplice analisi sull'età degli input di ciascun TXO può portare a indovinare il TXO realmente speso con una probabilità maggiore di $\frac{1}{l}$. In particolare per dimostrare ciò sono state estratte informazioni rilevanti dalla blockchain di Monero, fino al blocco 1288774 (15 aprile 2017) e sono state memorizzate in un database a grafo. In seguito è stato sviluppato un algoritmo iterativo, nel quale, ad ogni iterazione, sono stati segnalati tutti i riferimenti ai mixin che non possono

rappresentare la spesa effettiva. Questo perché si era già determinato che l'output corrispondente era stato speso in una transazione diversa. Lo stesso è stato fatto per gli input reali tra ulteriori insiemi di input di transazione. Da questo è risultato che il 63% degli input delle transazioni Monero può essere tracciato grazie a questo approccio. E' stato dimostrato inoltre come la vulnerabilità delle transazioni Monero all'analisi deduttiva varia con il numero di mixin scelti dall'utente: le transazioni con più mixin sono significativamente meno deducibili, come ci si potrebbe aspettare. Inoltre, anche tra le transazioni con lo stesso numero di mixin, le transazioni effettuate con versioni successive del software sono meno vulnerabili.

Nonostante l'analisi sia stata condotta su Monero, poiché l'attacco di deducibilità deriva dalla procedura di campionamento dei mixin, intrinseca al protocollo Cryptonote, le altre criptovalute basate su di esso condividono le stesse vulnerabilità. Ad esempio, per Bytecoin, la prima implementazione del protocollo Cryptonote, si è dedotto l'input reale per il 29% degli input delle transazioni con 1 o più mixin. Sebbene questo tasso sia inferiore rispetto a Monero, si ipotizza che sia il risultato di un minor utilizzo di Bytecoin.

In un altro studio [**?**], gli autori hanno dimostrato che, nella maggior parte dei casi, il TXO più recente nel set di mixin è quello realmente speso. Hanno verificato la loro euristica mediante simulazione e hanno mostrato che il 99,5% degli input delle transazioni può essere tracciato utilizzando questo semplice attacco. Nonostante dall'aprile del 2015 gli sviluppatori di Monero sono passati a una distribuzione triangolare per selezionare i TXO precedenti come decoy, utilizzando lo stesso attacco, ancora il 96% delle transazioni può essere tracciato.

Seguendo il suggerimento proposto dagli autori di [**?**], Monero è poi passato alla distribuzione Gamma per la selezione dei decoy. L'uso di una distribuzione migliore, cioè una distribuzione più vicina alla distribuzione dell'età dei TXO realmente spesi, può rendere alcuni attacchi meno efficaci, come quello proposto in [**?**]. Tuttavia, poiché il comportamento degli utenti cambia nel tempo [**?**], è difficile trovare e seguire costantemente la distribuzione dell'età dei TXO realmente spesi e selezionare i decoy delle transazioni utilizzando esattamente la stessa distribuzione di età.

In [**?**] è stato proposto un ulteriore attacco, che sfrutta i record delle transazioni precedenti sulla blockchain per ottenere ipotesi probabilistiche sui TXO effettivamente spesi in una transazione. In particolare è stata costruita una transazione malevola per ridurre la k-anonimità o persino de-anonimizzare le transazioni Monero, con effetti simili al caso zero mixin.

Nell'articolo [**?**] viene descritto un attacco basato sull'idea che, dato che gli algoritmi di selezione dei mixin sono di tipo probabilistico, c'è la possibilità che alcuni TXO vengano involontariamente ignorati da altri utenti e non vengano utilizzati come mixin in nessuna transazione, ma piuttosto come TXO effettivamente spesi.

Supponendo che la rete approvi soltanto transazioni con anelli di dimensione pari a l. Sia $P_t(n)$ la probabilità che un nuovo TXO, X_N, venga trasferito quando esattamente n nuovi TXO sono registrati sulla blockchain dopo di esso. Sia $\epsilon(n)$ la probabilità che un TXO rimanga non speso dopo n TXO, la cui dipende dal comportamento degli utenti. In particolare, se una criptovaluta viene principalmente utilizzata come mezzo di scambio, dove gli utenti scambiano frequentemente la valuta per beni e servizi, allora $\epsilon(n)$ potrebbe essere molto vicino a zero per un n sufficientemente grande. Di seguito assumiamo che $\epsilon(n)$ abbia un valore trascurabile per grandi n al fine di rendere le equazioni più trattabili, tuttavia, gli attacchi proposti possono essere utilizzati per qualsiasi valore di $\epsilon(n)$. Sia $P_f(\cdot)$ è la distribuzione che l'algoritmo di selezione del mixin utilizza per selezionare i decoy e $P_0(K)$ è la probabilità che un TXO specifico rimanga ignorato per K transazioni, cioè non utilizzato come decoy nelle successive K transazioni, dopo essere stato emesso. Sia X_i l'i-esimo TXO sulla blockchain. Supponiamo, per ogni X_N, di esaminare gli insiemi di mixin di tutte le transazioni con output in $\{X_{N+1}, \ldots, X_{N+K}\}$. Se X_N compare nell'insieme di mixin di una sola transazione, allora è molto probabile che X_N sia il vero TXO speso in quella transazione. Se K è sufficientemente grande, allora il tasso di errore diminuisce e di conseguenza l'inferenza avrà una precisione maggiore in quanto i TXO sono spesi dall'utente dopo un tempo limitato. La Proposizione 1 fornisce la precisione di tale inferenza.

Proposizione 1: Sia X_N un output di una transazione che viene utilizzato, per la prima volta,come mixin esattamente k_1 transazioni dopo. Se X_N non viene utilizzato come mixin in nessun'altra transazione per almeno k_2 transazioni dopo la prima volta, allora la probabilità che la prima volta che è apparso sulla blockchain sia la sua vera spesa, indicata come $P(A|B_{k_1,k_2})$, è data da

$$P(A|B_{k_1,k_2}) = \frac{P_t(k_1)}{P_t(k_1) + \epsilon(k_1+k_2) \cdot \frac{1-(1-P_f(k_1))^{l-1}}{1-(1-P_f(k_1))^{l-1}}}. \quad (6)$$

dove gli eventi A e B_{k_1,k_2} sono definiti come:

- A è l'evento in cui X_N è speso nella k_1-esima transazione dopo la sua emissione;
- B_{k_1,k_2} è l'evento in cui X_N non viene utilizzato in nessuna delle $k_1 + k_2$ transazioni dopo la sua emissione, tranne per la k_1-esima.

Sia K il numero più piccolo per il quale $\epsilon(K)$ è trascurabile rispetto a $P_t(k_1)$ per tutti i k_1 minori di K. Allora, per tutti i k_1 e k_2 dove $k_1 + k_2 > K$, la probabilità in (6) è molto vicina a 1 e l'Attacco 1 avrà una precisione molto alta.

Supponendo che $\epsilon(n)$ diminuisca all'aumentare di n, la probabilità che una transazione selezionata casualmente venga tracciata quasi certamente utilizzando l'attacco, indicata con $P_r(Attacco1)$, è data da:

$$P_r(Attacco1) \approx P_0(K + 1). \tag{4.1}$$

È importante notare che, nonostante gli utenti di Monero utilizzino diversi codici sorgente, ciascuno con algoritmi differenti per la selezione dei mixin, questo tipo di attacco rimane applicabile, anche se valutare l'efficacia in questo contesto risulta complesso.

Capitolo 5

RandomX

5.1 Design di Randomx

Per minimizzare il vantaggio di hardware specializzati come gli ASIC, già precedentemente discussi, un algoritmo di **Proof of Work** (**PoW**) deve potersi legare ai dispositivi esistenti il cui uso risulti essere ampiamente diffuso. Non a caso, si focalizza sull'utilizzo delle **CPU** per i seguenti motivi:

- **Accessibilità**: Le CPU, essendo meno specializzate, sono più diffuse e accessibili. Un algoritmo basato su CPU è più **egalitario** e permette a più partecipanti di unirsi alla rete, contrariamente a ciò che accadeva con gli ASIC in quanto il loro notevole costo permetteva una centralizzazione del mining nelle mani di pochi;
- **Istruzioni Hardware Comuni**: CPU diverse **condividono** un grande *subset* di istruzioni hardware native;
- **Documentazione e Compilatori**: Tutti i principali set di istruzioni CPU sono ben **documentati** con diversi compilatori **open-source** disponibili.

5.1.1 Prova di Lavoro Dinamica

L'idea alla base di una **Proof of Work** basata su CPU è l'impiego di un *lavoro dinamico* sfruttando il fatto che queste accettano come input non solo **dati**, come le tipiche funzioni di hash crittografiche, ma anche il **codice**. Ciò evita che le sequenza delle operazioni sia fissa e quindi più facilmente eseguibile da un circuito integrato specializzato. Una **Proof of Work dinamica** consiste in 4 steps:

1. Generazione di un programma casuale;
2. Traduzione nel codice macchina nativo della CPU;
3. Esecuzione del programma;
4. Trasformazione dell'output del programma in un valore crittograficamente sicuro.

Generazione di un programma casuale

Inizialmente la progettazione di una Proof of Work si basava sulla generazione di un programma in linguaggi ad alto livello, come C o Javascript, ma questi per via della loro sintassi complessa, implicavano notevoli costi in termini di tempo.

Il modo più veloce per generare un programma casuale è utilizzare un generatore senza logica, riempiendo semplicemente un **buffer con dati casuali**. Questo richiede la progettazione di un linguaggio di programmazione senza sintassi, in cui tutte le stringhe di bit casuali rappresentano programmi validi.

Traduzione del Programma in Codice Macchina

Per generare il codice macchina il più velocemente possibile, il nostro set di istruzioni deve essere il più vicino possibile all'hardware nativo, ma allo stesso tempo deve risultare abbastanza generico in modo da non limitare l'algoritmo a una specifica architettura CPU.

Esecuzione del Programma

L'esecuzione del programma dovrebbe utilizzare il **maggior numero** possibile di **componenti della CPU**. Alcune delle caratteristiche che dovrebbero essere sfruttate sono:

- Cache multi-livello (L1, L2, L3);
- Cache µop;
- Unità logica aritmetica (ALU);
- Unità a virgola mobile (FPU);
- Controller di memoria;
- Parallelismo a livello di istruzione;
- Esecuzione fuori ordine;
- Esecuzione speculativa;
- Rinominazione dei registri.

Calcolo del Risultato Finale

Blake2b è una funzione di hashing crittograficamente sicura, progettata per essere veloce nel software, soprattutto sui moderni processori a 64 bit, dove è circa tre volte più veloce di SHA-3. Proprio per questo è ideale per essere utilizzata in una **Proof of Work** basata su **CPU**.

Per quanto riguarda, invece, l'elaborazione di grandi quantità di dati in modo crittograficamente sicuro, l'**Advanced Encryption Standard** (**AES**) può fornire la massima velocità di elaborazione perché molte CPU moderne supportano l'accelerazione hardware di queste operazioni.

5.1.2 "Easy program problem"

Il problema dell'*easy program* si basa sul fatto che quando viene generato un programma casuale, si potrebbe decidere di eseguirlo solo in caso sia favorevole. Questa strategia è fattibile per due motivi principali:

- **Distribuzione del Tempo di Esecuzione**: I tempi di esecuzione dei programmi generati casualmente seguono tipicamente una *distribuzione log-normale*. Questi possono rapidamente analizzati e in caso di tempo di esecuzione superiore alla media, l'esecuzione può essere saltata e può essere generato un nuovo programma. Se quest'ultima operazione risulta essere economica, può migliorare significativamente le prestazioni.
- **Ottimizzazione delle Caratteristiche**: Un'implementazione potrebbe scegliere di ottimizzare un sottoinsieme delle caratteristiche necessarie per l'esecuzione del programma. Ad esempio, si può decidere di eliminare il supporto per alcune operazioni (come la divisione) o di implementare alcune sequenze di istruzioni in modo più efficiente. In seguito, i programmi generati verrebbero analizzati ed eseguiti solo se soddisfano i requisiti specifici dell'implementazione ottimizzata.

Queste strategie di ricerca di programmi con particolari proprietà vanno in **contrasto** con gli obiettivi di questa **Proof of Work**, quindi devono essere eliminate:

- **Soluzione**: Esecuzione di una sequenza di **N programmi casuali**, in modo che ogni programma sia generato dall'**output del precedente**. L'output del programma finale viene quindi utilizzato come risultato.
- **Principio**: Una volta eseguito il primo programma, un miner deve decidere **se impegnarsi a terminare l'intera catena** (che può includere programmi sfavorevoli) o ricominciare da capo e sprecare lo sforzo impiegato sulla catena non completata;
- **Vantaggio**: Uniformare il tempo di esecuzione per l'intera catena, poiché la deviazione relativa di una somma di tempi di esecuzione distribuiti identicamente è ridotta.

5.1.3 Tempo di verifica

Poiché lo scopo del **Proof of Work** è di essere utilizzato in una **rete peer-to-peer** senza fiducia, i partecipanti alla rete devono essere in grado di verificare rapidamente se una prova è valida o meno. Ciò pone un **limite** superiore alla **complessità dell'algoritmo** di **Proof of Work**. In particolare, abbiamo fissato l'obiettivo per RandomX di essere almeno altrettanto veloce da verificare quanto la funzione di hash CryptoNight, che mira a sostituire.

5.1.4 Memory-hardness

Oltre alle risorse computazionali pure, come **ALU** e **FPU**, le **CPU** di solito hanno accesso a una grande quantità di memoria sotto forma di DRAM. Le prestazioni del sottosistema di memoria sono tipicamente ottimizzate per adattarsi alle capacità di calcolo.

Per utilizzare la **memoria esterna** così come i controller di memoria on-chip, l'**algoritmo di Proof of Work** dovrebbe accedere a un grande **buffer** di memoria (chiamato "**Dataset**"). Il Dataset deve essere:

- **più grande** di quanto possa essere memorizzato on-chip (per richiedere memoria esterna);
- **dinamico** (per richiedere memoria scrivibile)

Idealmente, la dimensione del Dataset dovrebbe essere di almeno 4 GiB. Tuttavia, a causa dei vincoli sul tempo di verifica la dimensione utilizzata da RandomX è stata selezionata a 2080 MiB.

Bibliografia

[1] Hyungmin Cho. Asic-resistance of multi-hash proof-of-work mechanisms for blockchain consensus protocols. *IEEE Access*, PP:1–1, 10 2018.

[2] Zonghao Feng and Qiong Luo. Evaluating memory-hard proof-of-work algorithms on three processors. *Proc. VLDB Endow.*, 13(6):898–911, feb 2020.