APPLICAZIONI DELLA TECNOLOGIA BLOCKCHAIN

VOLUME 1 - 2021

a cura di Danilo Bazzanella e Andrea Gangemi

Autori

Luca Ambrosino, Luca Bajardi, Simonetta Bodojra, Luca Cataldo, Giulio Cerruto, Alessio Claudio, Giorgio Colantoni, Giulia Corrente, Lorenzo De Siena, Jacopo Dominici, Rachid El Amrani, Martina Fraia, Carmen Frasca, Luca Giorgino, Alessandro Guggino, Enrico Guglielmino, Carlo Iurato, Adriano Koleci, Laura Marioni, Gianluca Mega, Marco Momo, Luca Montaldo, Matteo Pappadà, Alessandro Pino, Elena Pitino, Giulia Pititto, Daniele Plutino, Alessandro Rigoli, Luca Saglia, Tommaso Toso, Kairi Zuccarino.

DIPARTIMENTO DI SCIENZE MATEMATICHE
POLITECNICO DI TORINO

INDICE

MONERO

Luca Ambrosino, Luca Bajardi, Giorgio Colantoni, Martina Fraia

Indice

Capitolo 1

Introduzione

1.1 Monero

Figura 1.1: *Logo di Monero*

Monero (XMR) è una criptovaluta nata come fork di Bytecoin nel 2014, circa 5 anni dopo la prima e più famosa Bitcoin. Molti dei suoi creatori sono rimasti anonimi, ma alcune voci suggeriscono che dietro ci sia anche lo zampino dello stesso Satoshi Nakamoto, inventore di Bitcoin; naturalmente, nel corso degli anni molti sviluppatori hanno contribuito a far crescere monero fino al suo livello attuale.
Come le altre criptovalute, essa ha alcuni obiettivi ben precisi:

- La creazione di una moneta digitale **decentralizzata**, che abolisca dunque l'entità centrale delle banche per svolgere transazioni di ogni tipo;
- Garantire la **privacy** ai propri utenti in ogni transazione, indipendentemente dalle competenze tecnologiche di ciascuno di essi;
- Una sufficiente **scalabilità**, ovvero la capacità di crescere o diminuire in base alla necessità.

In particolare, Monero vuole cercare soluzioni alternative a Bitcoin in quanto si ritiene che la privacy di quest'ultima non sia del tutto solida. Per fare questo, inizialmente Monero si basava sul protocollo *CryptoNight*, derivato dall'algoritmo *CryptoNote* [25]; tuttavia, con l'avvento degli ASICs (*Application Specific Integrated Circuits*) in grado di eseguire le PoW del protocollo *CryptoNight* in un tempo molto minore rispetto a un dispositivo comune, nel novembre 2019 la rete di Monero ha deciso di passare al protocollo **RandomX**.

Uno dei principali obiettivi del progetto Monero è raggiungere il massimo livello di **decentralizzazione** possibile, il che significa che un utente non ha bisogno di fidarsi di nessun altro sulla rete al di fuori di se stesso.

Un altro aspetto fondamentale che contraddistingue Monero da altre criptovalute è la speciale attenzione che rivolge alla **privacy**: mentre ogni Bitcoin in circolazione ha un proprio "numero di serie" che lo rende monitorabile, XMR è completamente fungibile. I dettagli sui mittenti e sui destinatari, così come la quantità di criptovaluta trasferita sono oscurati e i sostenitori di Monero affermano che questa caratteristica offre un vantaggio a discapito di altre criptomonete focalizzate sulla privacy come ad esempio *Zcash*.

La privacy in Monero viene garantita sotto diversi aspetti:

- Per il mittente dalla **ring signature**;
- Per il destinatario dagli *stealth addresses*, noti anche come **one-time addresses**;
- L'ammontare della transazione è protetto grazie alla **ring confidential transaction**;
- L'Internet Protocol Address (IP) degli utenti, che identifica univocamente un dispositivo collegato ad una rete informatica, è protetto da **Kovri**.

Nelle *ring signatures* gli output delle transazioni passate vengono prelevati dalla blockchain e utilizzati da esche, in modo che gli osservatori esterni non possano dedurre chi sia il proprietario della firma; ipotizzando poi che Alice voglia inviare 200 XMR a Bob, questo importo potrebbe anche essere suddiviso in blocchi casuali per aggiungere un ulteriore livello di difficoltà. Per garantire inoltre che le transazioni non possano essere collegate tra loro, vengono generati indirizzi nascosti per ogni singola transazione che vengono utilizzati solo una volta, i cosiddetti *one-time addresses*.
Il lato negativo di tutti questi vantaggi sulla privacy è chiaro: eventuali malfattori potrebbero approfittarsene; molti governi di tutto il mondo hanno infatti offerto centinaia di migliaia di dollari a chiunque riesca a decifrare il codice di Monero per smascherare operazioni illecite.

1.2 Monero sul mercato

Il valore di Monero [10] ha recentemente avuto il suo massimo il 7 Maggio 2021 toccando quota 419,54€; attualmente (27/05/2021) invece 1 Monero vale 211,67€, ha una capitalizzazione di mercato di 3.792.869.270,62€ (quasi 4 miliardi) il che la rende la 27-esima criptovaluta al mondo, grazie alla circolazione di circa 17,7 milioni di Monero. Come per Bitcoin, anche per Monero è fissato un tetto massimo di unità minabili, pari a 18,4 milioni, il quale si stima verrà raggiunto il 31 maggio 2022, cioè tra circa un anno.

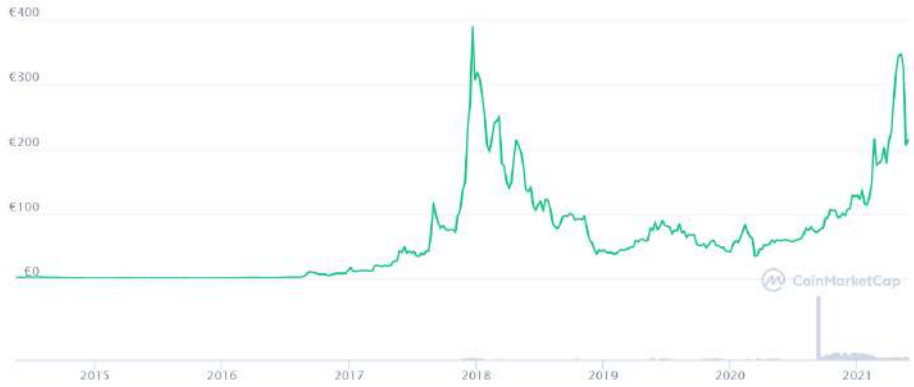


Figura 1.2: *Andamento di Monero - All time* [10]

Figura 1.3: *Andamento di Monero - Ultimo anno* [10]

1.3 Blocchi in Monero

Per i primi due anni la **velocità** di inserimento dei blocchi era di 1 al minuto, ma dopo il 2016 è raddoppiata per stabilizzarsi sugli attuali 2 minuti, e la **difficoltà** per minare si autoregola dopo ogni blocco in modo da mantenere questa media. Tuttavia a differenza di Bitcoin, in Monero sia la **dimensione** di ogni blocco sia le **ricompense** per i miners sono dinamiche e variano da uno all'altro.

1.3.1 Block reward

In generale è fondamentale che i miners ricevano delle ricompense per minare i blocchi, altrimenti perderebbero interesse nella blockchain, il che renderebbe più probabile che un hacker malevolo possa controllare più del 50% della potenza di calcolo della rete e sia quindi in grado di modificare sostanzialmente la catena a suo piacimento.

Per calcolare le ricompense che spettano ai miners [13], si usa un operatore di shift: l'operatore $n >> k$ shifta di k posizioni verso destra il numero n scritto in base 2 (ovvero divide il numero in base 10 per 2^k). Supponiamo di avere il numero $n = 10$ che in binario è $[1010]_2$; scrivere $n >> 2$ significa shiftare verso destra di 2 posizioni il numero $[1010]_2$, ottenendo così $[0010.10]_2$, cioè 2.5 in base dieci[1].
Dati M il totale di XMR nella rete e $L = 2^{64} - 1$ il tetto massimo, la ricompensa B, in unità atomiche, cioè $1XMR = 10^{12} u.a.$, per l'inserimento del blocco veniva inizialmente calcolata come

$$B = [L - M]_2 >> 20$$

Tuttavia, dopo il cambiamento del tempo di inserimento da 1 a 2 minuti avvenuto nel 2016, per equità sono raddoppiate le ricompense diventando così:

$$B = [L - M]_2 >> 19.$$

Quando nel maggio 2022 sarà raggiunto il tetto massimo di 18,4 milioni di Monero in circolazione, tecnicamente i miners non dovrebbero ricevere più alcuna ricompensa. Questo però sarebbe un problema per la rete in quanto se calasse l'interesse dei miners nell'inserire nuovi blocchi, si rischierebbe il collasso totale. Per prevenire questo problema, i miners saranno incentivati utilizzando delle "*emissioni di coda*", ovvero una piccola quantità di Monero (si pensa 0.3 XMR) immessa nel sistema ogni 60 secondi come ricompensa, in modo che la ricompensa per l'inserimento di un blocco sia di 0.6 XMR; così si rischierà da un lato una leggera inflazione, ma dall'altro si terrà vivo l'interesse dei miners nella criptomoneta.

[1] In realtà le cifre dopo la virgola vengono scartate, ottenendo quindi $[0010]_2$, cioè 2.

1.3.2 Dimensione dei Blocchi

L'introduzione di blocchi di dimensione dinamica ha sicuramente alcuni vantaggi e alcuni svantaggi:

- **Pro**: ottima *scalabilità*, infatti l'assenza di un "tetto" permette di aumentare notevolmente il volume di transazioni, inserendo nei blocchi anche quelle con fees piccole che risulterebbero altrimenti penalizzate da blocchi con dimensione massima fissata; queste ultime infatti non dovranno aspettare magari delle ore come in Bitcoin per essere immesse nel sistema e approvate, ma basteranno pochissimi minuti.
- **Contro**: il fatto di non avere una dimensione massima rischia di appesantire troppo la catena nel caso in cui i miners cominciassero a inserire blocchi troppo grandi; come vedremo, questo problema viene in parte risolto grazie all'inserimento di *penalità* verso i miners che inseriscono blocchi troppo grandi.

Adesso discutiamo delle componenti che determinano il *weight* di una transazione e successivamente dell'intero blocco. Il peso `transaction_weight` di una transazione si calcola come:

$$\texttt{transaction\_weight} = \texttt{transaction\_size} + \texttt{transaction\_clawback}$$

dove `transaction_size` è la dimensione in bytes della transazione, mentre `transaction_clawback` è un costo in spazio aggiuntivo dovuto alla verifica sulla correttezza degli *amount* effettuata dal protocollo Bulletproof, durante il Range Proof. Definiamo quindi `block_weight` la somma di tutti i `transaction_weights` e del peso della transazione di reward del miner.

Associamo poi ad ogni blocco un altro valore chiamato `longterm_block_weight`: il motivo della sua esistenza risiede nel fatto che con i blocchi dinamici la catena rischia di diventare dimensionalmente ingestibile. Per calcolare quest'ultimo bisogna introdurre il concetto di `effective_longterm_median`, che tiene conto della mediana dei pesi dei 10000 blocchi precedenti, compreso il blocco stesso. Calcoliamo quindi il `long_term_block_weight` come il minimo tra il `block_weight` e l'`effective_longterm_median` moltiplicato di un fattore 1.4 del blocco precedente:

$$\texttt{longterm\_block\_weight} = \min\{\texttt{block\_weight}, 1.4 \cdot \texttt{previous\_effective\_longterm\_median}\}$$

Questi valori ci danno indicazioni sull'evoluzione delle dimensioni dei blocchi per tempi lunghi, ad esempio affinché la dimensione media di un blocco aumenti di un 40% si stima che servano almeno 50'000 blocchi della dimensione maggiorata. Monero offre ai miners anche una flessibilità a breve termine per la dimensione di un blocco. Questa può essere utile in quanto si possono verificare dei periodi in cui si ha una drastica variazione dei volumi delle transazioni nella blockchain. In tal caso si calcola il valore `cumulative_weights_median`, che tiene conto

della mediana dei pesi degli ultimi 100 blocchi, e l'`effective_longterm_median` dell'ultimo blocco inserito. Infine è possibile calcolare il vincolo che stabilisce il peso massimo del blocco che si può inserire, il quale risulta essere:

$$\texttt{max\_next\_block\_weight} = 2 \cdot \texttt{cumulative\_weights\_median}$$

Per evitare che i miners attacchino blocchi sempre più grossi, cioè della grandezza di `max_next_block_weight`, il protocollo di Monero ha introdotto due sistemi:

1. Una **penalità** sulla ricompensa dei miners qualora il blocco superi il peso `cumulative_weights_median`. L'ammontare di questa penalità dipende da quanto si sfora, cioè da quanto la grandezza del blocco inserito supera la precedente `cumulative_weights_median`. Infatti, tale penalità P è calcolata come:

   $$P = B \cdot \left(\frac{\texttt{block\_weight}}{\texttt{cumulative\_weights\_median}} - 1 \right)^2$$

   dove B e il reward del blocco. Dunque il miner riceve $B_P = B - P$, cioè:

   $$B_P = B \cdot \left(1 - \left(\frac{\texttt{block\_weight}}{\texttt{cumulative\_weights\_median}} - 1 \right)^2 \right)$$

   l'operatore di elevamento al quadrato fa sì che la penalità sia sub-proporzionale rispetto al peso del blocco.

2. Il secondo meccanismo adottato è stato quello di rendere dinamico anche il valore minimo di fee che può costare una transazione. Questa modifica è stata introdotta per impedire che i miners potessero compensare la penalità con solo il valore delle fees delle transazioni, ricevendo così la totalità del reward. Il sistema è abbastanza complicato, dunque non sarà trattato nello specifico in questo documento, ma si possono trovare alcuni riferimenti in [7, 8, 9, 12].

1.4 Monero mining e Monero PoW

1.4.1 Monero proof of work (PoW)

Monero ha utilizzato diversi algoritmi di hash per la proof of work (con output a 32 byte) in diverse versioni di protocollo. L'originale, noto come *Cryptonight*, è stato progettato per essere relativamente inefficiente su GPU, FPGA e architetture ASICs rispetto alle funzioni hash standard, come SHA256. Nell'aprile 2018 (v7 del protocollo), sono stati necessari nuovi blocchi per iniziare a utilizzare una variante leggermente modificata che ha contrastato l'avvento degli *ASIC Cryptonight* [16]. Un'altra leggera variante, denominata *Cryptonight V2*, è stata implementata nell'ottobre 2018 (v8) [17] e *Cryptonight-R* (basato su

Cryptonight ma con modifiche più sostanziali di un semplice ritocco) ha iniziato a essere utilizzato per i nuovi blocchi a marzo 2019 (v10) [18]. Una nuova proof of work che si distanziò radicalmente dalle precedenti si ebbe con *RandomX* [27], la quale è stata progettata e resa obbligatoria per i nuovi blocchi nel novembre 2019 (v12) con l'intenzione di una resistenza agli ASICs a lungo termine [19]. Quest'ultima è assicurata, temporaneamente, da algoritmi dinamici che non hanno un flusso di esecuzione statico, in quanto risulta essere molto più semplice cablare un insieme fisso di operazioni in un chip. L'idea di base di *RandomX* è che ogni stringa di 8 bits generata casualmente sia un'istruzione valida e l'insieme di istruzioni compongono un programma in grado di eseguire la PoW. Inoltre gli sviluppatori di *RandomX* hanno deciso di puntare molto sulla *CPU*, la quale è in grado di eseguire una grande varietà di codici, rendendo la costruzione di un ASIC equivalente alla costruzione di una CPU. Un secondo motivo che ha portato a orientarsi verso le CPUs è che quest'ultime sono molto più accessibili rispetto alle GPUs e hanno anche un mercato molto più ampio e in sviluppo negli ultimi anni, con una possibile crescita del bacino d'utenza.

1.4.2 Monero mining

Solitamente per calcolare la velocità di mining vengono presi in considerazione un gruppo di b blocchi recenti dalla blockchain, che indicheremo con l'indice $u \in 1, \ldots, b$. Ciascuno di questi blocchi avrà una difficoltà d_u. Ipotizziamo che i nodi che hanno minato questi blocchi siano onesti e che quindi la timestamp TS_u sia precisa, allora il tempo totale tra il primo blocco e il più recente è definito come: $\texttt{totalTime} = TS_b - TS_1$. Il numero adatto di hash richieste per minare tutti i blocchi è $\texttt{totalDifficulty} = \sum_u d_u$. Dunque ora è possibile calcolare quanto è veloce la rete, con tutti i suoi nodi, a calcolare le hash. Se la velocità attuale non cambia molto nel tempo necessario per produrre tutti i b blocchi, abbiamo: $\texttt{hashSpeed} \approx \texttt{totalDifficulty}/\texttt{totalTime}$.
É possibile fissare un target time per minare nuovi blocchi così che i blocchi vengano prodotti ad un rate pari a un blocco ogni target time, quindi è possibile calcolare quante hash deve fare la rete nel target time alla velocità calcolata sopra e si ottiene:

$$\texttt{newDifficulty} = \texttt{hashSpeed} \cdot \texttt{targetTime}$$

Non c'è alcuna garanzia che il prossimo blocco prenda la `newDifficulty`, ma nel tempo e dopo molti blocchi, continuando a calibrare, la difficoltà andrà come la vera `hashSpeed` della rete e i blocchi tenderanno al `targetTime`.

Monero per assicurarsi che le fork della catena siano su un piano di parità, non considera i blocchi più recenti per calcolare le nuove difficoltà, ma ritarda il nostro gruppo b di l. Ad esempio, se ci sono 39 blocchi nella catena (blocchi $1, \ldots, 39$), $b = 15$ e $l = 10$, consideriamo i blocchi $15 - 29$ per calcolare la difficoltà del blocco 40.

Se i nodi di mining sono disonesti, possono manipolare i timestamp in modo che le nuove difficoltà non corrispondano alla reale velocità di hash della rete. Risolviamo questo problema ordinando cronologicamente i timestamp, quindi tagliando i primi o outliers e gli ultimi o outliers. Ora abbiamo una "finestra" di blocchi $w = b - 2 \cdot o$. Dall'esempio precedente, se $o = 4$ e i timestamp sono onesti, taglieremmo i blocchi $15 - 18$ e $26 - 29$, lasciando i blocchi $19 - 25$ da cui calcolare la difficoltà del blocco 40.

Prima di escludere gli outliers abbiamo ordinato i timestamp, ma *solo* i timestamp. Le difficoltà dei blocchi non vengono ordinate. Usiamo la difficoltà cumulativa per ogni blocco, che è la difficoltà di quel blocco più la difficoltà di tutti i blocchi precedenti nella catena.

Usando gli array (tagliati) di w timestamp ordinati e difficoltà cumulative non ordinate (indicizzate da $1, \ldots, w$), definiamo

$\texttt{totalTime} = \texttt{choppedSortedTimestamps}[w] - \texttt{choppedSortedTimestamps}[1]$

$\texttt{totalDifficulty} = \texttt{choppedCumulativeDifficulties}[w] - \texttt{choppedCumulativeDifficulties}[1]$

In Monero il tempo target è 120 secondi (2 minuti)[2], $l = 15$ (30 minuti), $b = 720$ (un giorno) e $o = 60$ (2 ore).

Le difficoltà dei blocchi non sono memorizzate nella blockchain, quindi qualcuno che scarica una copia della blockchain e verifica che tutti i blocchi siano validi, necessita di ricalcolare le difficoltà dai timestamp registrati. Ci sono alcune regole da considerare per i primi $b + l = 735$ blocchi.

> **Regola 1**: ignora completamente il blocco di genesi (blocco 0, con $d = 1$). I blocchi 1 e 2 hanno $d = 1$.
>
> **Regola 2**: prima di eliminare gli outliers, provare a ottenere la finestra w da cui calcolare i totali.
>
> **Regola 3**: dopo w blocchi, elimina gli outlier (sia quelli maggiori sia quelli minori), scalando la quantità tagliata fino a b blocchi. Se la quantità di blocchi precedenti (meno w) è dispari, rimuovi un valore anomalo più basso rispetto a quello alto.
>
> **Regola 4**: Dopo b blocchi, campiona i primi blocchi b fino ai blocchi $b+l$, dopodiché tutto procede normalmente - in ritardo di l.

1.5 Fork di Monero

Monero non è perfettamente resistente agli ASICs, ma il costo di crearne uno appositamente per Monero sarebbe troppo alto e non ne varrebbe la pena [2].

[2]Come detto prima, nel marzo 2016 (v2 del protocollo), Monero ha cambiato il tempo target da 1 a 2 minuti [24]

Questo perchè appena una società di hardware mining riesce a creare un ASIC che riesce a minare Monero, quest'ultimo aggiorna il suo protocollo rendendo così inutili gli hardware specializzati al mining di Monero.
Come spesso è accaduto anche per altre criptovalute, questi aggiornamenti di protocollo hanno causato nel tempo delle **hard fork** da cui sono nate nuove criptomonete (vedi la Figura 1.4), alcune con lo scopo di rendere invece possibile minare Monero servendosi di ASIC.

Figura 1.4: *Simboli delle principali fork di Monero* [2]

Capitolo 2

Concetti di base

2.1 La curva ellittica

Tipicamente, le criptomonete (ad esempio Bitcoin) utilizzano curve ellittiche che soddisfano l'*equazione di Weierstrass*:

$$y^2 = x^3 + ax + b \quad \text{con} \quad a, b, x, y \in \mathbb{F}_p.$$

La criptomoneta Monero, invece, utilizza una curva speciale che offre una sicurezza superiore rispetto alle altre *curve NIST* e un'eccellente performance delle primitive crittografiche. Essa appartiene alla categoria delle *Twisted Edwards curves* [4], che sono espresse come:

$$ax^2 + y^2 = 1 + dx^2y^2 \quad \text{con} \quad a, d, x, y \in \mathbb{F}_p. \tag{2.1}$$

Un vantaggio rispetto alle curve nella forma di Weierstrass è che solitamente richiedono meno operazioni aritmetiche. Monero utilizza una particolare *Twisted Edwards elliptic curve* per le operazioni crittografiche: **Ed25519** [3, 4, 5]. Essa è definita sul campo $\mathbb{F}_{2^{255}-19}$ attraverso la seguente equazione:

$$-x^2 + y^2 = 1 - \frac{121665}{121666}x^2y^2 \tag{2.2}$$

Le curve twisted Edwards hanno ordine esprimibile come $2^c \cdot q$ dove q è un numero primo e c un intero positivo. Nel caso della curva *Ed25519*, il suo ordine è un numero con 76 cifre decimali:

$2^3 \cdot 7237005577332262213973186563042994240857116359379907606001950938285454250989$

2.1.1 Compressione della chiave

Gli elementi del campo sono interi di 256-bit, quindi possono essere rappresentati usando 32 bytes, così che ogni punto della curva può essere rappresentato usando 64 bytes. La curva *Ed25519* permette di applicare facilmente la tecnica

di compressione dei punti dimezzando il numero di bytes e salvando solo una coordinata mettendo il bit più significativo della coordinata y a 0 se x è pari e 1 se è dispari. Il valore di y rappresenterà il punto della curva.

Per riottenere il punto nel formato (x, y):

1. Recupera e cancella il bit b più significativo del valore memorizzato, e quello che rimane sarà y.
2. Calcola $u = y^2 - 1 \pmod p$ e $v = dy^2 + 1 \pmod p$ con $d = \frac{121665}{121666}$. Questo significa $x^2 = u/v \pmod p$.
3. Calcola[1] $z = uv^3(uv^7)^{(p-5)/8} \pmod p$.
4. Se $vz^2 = u \pmod p$, allora $x' = z$.
5. Altrimenti calcola $x' = z \cdot 2^{(p-1)/4} \pmod p$.
6. Utilizza il bit di parità b recuperato nel primo passaggio, se $b \neq$ dal bit meno significativo di x', allora restituisci $x = -x'$, altrimenti restituisci $x = x'$.
7. Restituisci il punto decompresso (x, y).

2.1.2 Algoritmo della firma EdDSA

Bernestein e il suo team proposero [3, 5] un'alternativa molto ottimizzata e sicura della firma ECDSA che permette di produrre più di 100'000 firme al secondo usando un processore *Intel Xeon*. Invece di generare ogni volta numeri casuali viene usata una hash di un valore derivato dalla chiave privata di chi firma. Questo permette di evitare tutti i difetti collegati all'implementazione di generatori di numeri casuali. Un altro obiettivo è quello di evitare di accedere a posizioni di memoria segrete o imprevedibili per prevenire i cosiddetti attacchi *cache timing*.

Firma

1. Sia h_k un hash $\mathcal{H}(k)$ della chiave privata k del firmatario. Calcola r come un hash $r = \mathcal{H}(h_k, \mathfrak{m})$ dell'hash della chiave privata e del messaggio.
2. Calcola[2] $R = rG$ e $s = (r + \mathcal{H}(R, K, \mathfrak{m}) \cdot k))$
3. La firma è la coppia (R, s)

Di default, una firma *EdDSA* richiederebbe 64 + 32 byte (64 per R e 32 per s) per essere rappresentata. Assumendo che il punto R sia compresso, allora i requisiti di spazio vengono ridotti a 32 + 32 byte.

Verifica
La verifica viene eseguita come segue:

[1] Visto che $p = 2^{255} - 19 \equiv 5 \pmod 8$, $(p-5)/8$ e $(p-1)/4$ sono interi.
[2] Il generatore della curva è il punto $G = (x, 4/5)$ per il quale x è positivo.

1. Calcola $h = \mathcal{H}(R, K, \mathfrak{m})$
2. Se l'uguaglianza $(2^c s)G = 2^c R + 2^c \mathcal{H}(R, K, \mathfrak{m})K$ vale, allora la firma è valida

Correttezza
Il motivo per cui la verifica della firma funziona può essere derivato dalla seguente uguaglianza:

$$2^c sG = 2^c((r + \mathcal{H}(R, K, \mathfrak{m}) \cdot k) \cdot G = 2^c R + 2^c \mathcal{H}(R, K, \mathfrak{m}) \cdot K)$$

Capitolo 3

Ring signatures

Le **Ring signatures** sono firme generate con una singola chiave privata e un insieme di chiavi pubbliche non collegate. L'intero insieme di chiavi pubbliche, inclusa quella corrispondente alla chiave privata, è chiamato *ring*.
Inizialmente erano chiamate *Group signatures* poiché erano viste come un modo per dimostrare che una persona che firma apparteneva a un gruppo senza doversi necessariamente identificare come individuo. Nelle transazioni Monero, questo aiuta a rendere il flusso di denaro non tracciabile.
Le ring signatures hanno delle proprietà che rendono le transazioni confidenziali:

- **Anonymity** (anonimità): ogni osservatore non può essere in grado di identificare chi firma ma solo che la chiave privata utilizzata corrisponde a una delle chiavi pubbliche nel ring.

- **Linkability** (collegabilità): se una chiave privata è stata usata per firmare due messaggi diversi, allora i due messaggi diventano collegati e la duplicità non viene coperta. Questo aiuta a prevenire il double spending.

- **Exculpability** (non colpevolezza): se la chiave pubblica di un membro del ring è stata usata due volte in 2 firme diverse ma non ha firmato entrambe i due messaggi non vengono collegati.

Liu et al. [14] crearono un algoritmo per *Linkable and Spontaneous group signatures* che non richiede la collaborazione di co-firmatari. Inizialmente le group signatures richiedevano un gruppo di persone che si fidassero e gestissero delle firme collettive, che poteva possibilmente escludere il firmatario originale.

3.1 Linkable Spontaneous Anonymous Group Signatures (LSAG)

Firma
Sia $\mathfrak{m}$ il messaggio da firmare, $\mathcal{R} = \{K_1, K_2, \ldots, K_n\}$ un insieme di chiavi

pubbliche distinte (*group/ring*), k_π la chiave privata corrispondente a $K_\pi \in \mathcal{R}$. Assumiamo anche l'esistenza di due funzioni hash $\mathcal{H}_n$ e $\mathcal{H}_p$, che mappano rispettivamente in numeri interi (mod N) e punti della curva.[1]

1. Calcola $\widetilde{K} = k_\pi \mathcal{H}_p(\mathcal{R})$
2. Genera numeri casuali $\alpha \in \mathbb{Z}_q$ e $r_i \in \mathbb{Z}_q$ per $i \in \{0, 1, \ldots, n\}$ e $i \neq \pi$
3. Calcola
$$c_{\pi+1} = \mathcal{H}_n(\mathcal{R}||\widetilde{K}||\mathfrak{m}||\alpha G||\alpha\mathcal{H}_p(\mathcal{R}))$$
4. Per $i = \pi+1, \pi+2, \ldots, n, 1, 2, \ldots, \pi-1$ calcola, sostituendo $n+1 \to 1$
$$c_{i+1} = \mathcal{H}_n(\mathcal{R}||\widetilde{K}||\mathfrak{m}||r_i G + c_i K_i||r_i\mathcal{H}_p(\mathcal{R}) + c_i\widetilde{K})$$
5. Definisce $r_\pi = \alpha - k_\pi c_\pi \pmod q$

La firma sarà
$$\sigma(\mathfrak{m}) = (c_1, r_1, \ldots, r_n, \widetilde{K}).$$

Verifica
La verifica di una firma avviene nel modo seguente:

1. Per $i = 1, 2, \ldots, n$ calcolare, sostituendo $n+1 \to 1$:
$$\begin{aligned} z_i' &= r_i G + c_i K_i \\ z_i'' &= r_i \mathcal{H}_p(\mathcal{R}) + c_i \widetilde{K} \\ c_{i+1}' &= \mathcal{H}_n(\mathcal{R}||\widetilde{K}||\mathfrak{m}||z_i'||z_i'') \end{aligned}$$
2. se $c_1' = c_1$ allora la firma è valida

Correttezza
Possiamo convincerci che l'algoritmo funziona osservando quanto segue:

Se $i \neq \pi$ allora c_{i+1}' è definito come nell'algoritmo di firma.

Se $i = \pi$ allora

$$\begin{aligned} z_i' &= r_i G + c_i K_i = (\alpha - k_\pi c_\pi)\, G + c_\pi K_\pi = \alpha G \\ z_i'' &= r_i \mathcal{H}_p(\mathcal{R}) + c_i \widetilde{K} = (\alpha - k_\pi c_\pi)\, \mathcal{H}_p(\mathcal{R}) + c_\pi k_\pi \mathcal{H}_p(\mathcal{R}) = \alpha \mathcal{H}_p(\mathcal{R}) \end{aligned}$$

Quindi anche in questo caso l'espressione $c_{i+1}' = \mathcal{H}_n(\mathcal{R}||\widetilde{K}||\mathfrak{m}||z_i'||z_i'')$ sarà uguale a c_{i+1}.

[1] Una semplice definizione di $\mathcal{H}_p$ potrebbe essere $\mathcal{H}_p(x) = \mathcal{H}_n(x)G$

Linkability
Dato un set fisso di chiavi pubbliche R e due firme valide per messaggi diversi: $\sigma = (c_1, s_1, ..., s_n, \widetilde{K})$, $\sigma' = (c'_1, s'_1, ..., s'_n, \widetilde{K}')$.
Se $\widetilde{K} = \widetilde{K}'$ chiaramente entrambe le firme provengono dallo stesso anello di firma e dalla stessa chiave privata. In altre parole, lo schema di firma LSAG produce firme reciprocamente collegabili nel caso in cui un anello e una chiave privata vengano riutilizzati.

Non colpevolezza
Allo stesso tempo, dato che $\widetilde{K} = k_\pi \mathcal{H}_p(\mathcal{R})$, possiamo facilmente vedere che la collegabilità si applicherebbe solo se la chiave privata k_π fosse riutilizzata. Quindi, nessun altro membro del gruppo/anello potrebbe essere accusato di aver firmato due volte.

3.2 Back Linkable Spontaneous Anonymous Group Signatures (bLSAG)

Nella LSAG la collegabilità è garantita solo se il gruppo/ring è costante. Ciò è dovuto al fatto che $\widetilde{K}$ è definita attraverso $\mathcal{R}$. In questa versione viene garantita la *linkability* della chiave privata usata anche permettendo al ring di contenere chiavi diverse. L'idea principale è quella di calcolare $\widetilde{K}$ non tramite l'hash di $\mathcal{R}$ ma della chiave pubblica di chi firma e di rimuovere $\mathcal{R}$ nel calcolo di c_i.

Firma

1. Calcola $\widetilde{K} = k_\pi \mathcal{H}_p(K_\pi)$
2. Genera numeri casuali $\alpha \in \mathbb{Z}_q$ e $r_i \in \mathbb{Z}_q$ per $i \in \{0, 1, \ldots, n\}$ con $i \neq \pi$
3. Calcola
$$c_{\pi+1} = \mathcal{H}_n(\mathfrak{m}||\alpha G||\alpha \mathcal{H}_p(K_\pi))$$
4. Per $i = \pi + 1, \pi + 2, \ldots, n, 1, 2, \ldots, \pi - 1$ calcola, sostituendo $n + 1 \to 1$
$$c_{i+1} = \mathcal{H}_n(\mathfrak{m}||r_i G + c_i K_i||r_i \mathcal{H}_p(K_i) + c_i \widetilde{K})$$
5. Definisce $r_\pi = \alpha - k_\pi c_\pi \pmod q$

La firma sarà
$$\sigma(\mathfrak{m}) = (c_1, r_1, \ldots, r_n, \widetilde{K}).$$

Con questo tipo di firma due messaggi sono collegabili se e solo se la stessa chiave pubblica viene usata in entrambi, indipendentemente dal ring utilizzato da chi firma. Per Monero questa nozione di linkability è più utile di quella offerta dell'algoritmo LSAG perchè permette di controllare il double spending senza porre limiti sui membri usati per il ring.

3.3 Multilayer Linkable Spontaneous Anonymous Group Signatures (MLSAG)

Noether S. et al. [21, 22] descrissero una generalizzazione multi-layer della firma bLSAG applicabile quando abbiamo un set di $n \cdot m$ chiavi, dove m indica il numero di chiavi private con le quali si vuole firmare. Consideriamo il set $\mathcal{R} = \{K_{i,j}\}$ per $i \in 1, \ldots, n$ e $j \in 1, \ldots, m$, per la quale noi conosciamo le chiavi private $\{k_{\pi,j}\}$ corrispondenti alle chiavi pubbliche $\{K_{\pi,j}\}$ per qualche indice π. É necessario dunque generalizzare la nozione di linkability.

Linkability: Se qualcuna delle chiavi private $k_{\pi,j}$ viene usata per due diverse firme allora le due transazioni vengono collegate.

Firma

1. Calcola $\widetilde{K}_j = k_{\pi,j}\mathcal{H}_p(K_{\pi,j})$ per $j \in \{1, \ldots, m\}$

2. Genera numeri casuali $\alpha_j \in \mathbb{Z}_q$ e $r_{i,j} \in \mathbb{Z}_q$ (escluso $r_{\pi,j}$) per $j \in \{1, \ldots, m\}$ e $i \in \{0, 1, \ldots, n\}$ con $i \neq \pi$

3. Calcola
$$\begin{aligned} c_{\pi+1} = \mathcal{H}_n(\mathfrak{m}|| \\ & \alpha_1 G||\alpha_1 \mathcal{H}_p(K_{\pi,1})|| \\ & \ldots || \\ & \alpha_m G||\alpha_m \mathcal{H}_p(K_{\pi,m})) \end{aligned}$$

4. Per $i = \pi + 1, \pi + 2, \ldots, n, 1, 2, \ldots, \pi - 1$ calcola, sostituendo $n + 1 \to 1$
$$\begin{aligned} c_{i+1} = \mathcal{H}_n(\mathfrak{m}|| \\ & r_{i,1}G + c_i K_{i,1}||r_{i,1}\mathcal{H}_p(K_{i,1}) + c_i \widetilde{K}_1|| \\ & \ldots || \\ & r_{i,m}G + c_i K_{i,m}||r_{i,m}\mathcal{H}_p(K_{i,m}) + c_i \widetilde{K}_m) \end{aligned}$$

5. Definisce $r_{\pi,j} = \alpha_j - k_{\pi,j}c_\pi \pmod q$

La firma sarà

$$\sigma(\mathfrak{m}) = (c_1, r_{1,1}, \ldots r_{n,1}, \ldots, r_{1,m}, \ldots, r_{n,m}, \widetilde{K}_1, \ldots, \widetilde{K}_m).$$

Verifica
La verifica di una firma avviene nel modo seguente:

1. Per $i = 1, 2, \ldots, n$ calcolare, sostituendo $n + 1 \to 1$:

$$\begin{aligned} c'_{i+1} = \mathcal{H}_n(\mathfrak{m}|| \\ & r_{i,1}G + c_i K_{i,1}|| \quad r_{i,1}\mathcal{H}_p(K_{i,1}) + c_i \widetilde{K}_1|| \\ & \ldots || \\ & r_{i,m}G + c_i K_{i,m}|| \quad r_{i,m}\mathcal{H}_p(K_{i,m}) + c_i \widetilde{K}_m) \end{aligned}$$

2. se $c'_1 = c_1$ allora la firma è valida

Linkability
Come per la firma LSAG, possiamo osservare immediatamente che:

se $i \neq \pi$ allora c'_{i+1} è definito come nell'algoritmo di firma.

se $i = \pi$ allora, dato che $r_{\pi,j} = \alpha_j - k_{\pi,j}c_\pi$, vale:

$$\begin{aligned} r_{\pi,j}G + c_\pi K_{\pi,j} &= (\alpha_j - k_{\pi,j}c_\pi)\, G + c_\pi K_{\pi,j} = \alpha_j G \\ r_{\pi,j}\mathcal{H}_p(K_{\pi,j}) + c_\pi \widetilde{K}_j &= (\alpha_j - k_{\pi,j}c_\pi)\, \mathcal{H}_p(K_{\pi,j}) + c_\pi \widetilde{K}_j = \alpha_j \mathcal{H}_p(K_{\pi,j}) \end{aligned}$$

In altre parole $c'_{\pi+1} = c_{\pi+1}$.

Linkability
Nel caso in cui la chiave privata $k_{\pi,j}$ venisse riusata, allora il corrispondente valore di $\widetilde{K}_j$ nella firma lo rende evidente. Questa osservazione trova corrispondenza con la definizione generalizzata di connessione enunciata sopra.

Spazio richiesto
Assumendo la compressione dei punti, una firma MLSAG occupa un totale di $(1 + nm + m) \cdot 32$ bytes.

3.4 Borromean Ring Signatures

É necessario dimostrare che gli importi delle transazioni rientrano negli intervalli previsti, e questo può essere ottenuto utilizzando firme ad anello. Tuttavia, non è necessario che le firme siano collegabili, il che ci consente di selezionare algoritmi più efficienti in termini di spazio consumato.

In questo contesto, e allo scopo specifico di dimostrare intervalli di quantità, Monero utilizza uno schema di firma sviluppato da G. Maxwell [15]. Presentiamo qui una versione semplificata dello schema, in quanto supponiamo di avere lo stesso numero di chiavi per qualsiasi valore del primo indice i.
Nel nostro caso, le prove di intervallo richiederanno esattamente 2 chiavi per ogni cifra (questa semplificazione non avrà alcun impatto negativo).

Supponiamo di avere un insieme di chiavi pubbliche $\{K_{i,j}\}$ per $i \in \{1, 2, \ldots, n\}$ e $j \in \{1, 2, \ldots, m\}$. Inoltre, assumiamo che per ogni i ci sia un indice π_i tale che il firmatario conosca la chiave privata k_{i,π_i} corrispondente a K_{i,π_i}. Sia infine $\mathfrak{m}$ l'hash del messaggio concatenato con le chiavi $\{K_{i,j}\}$.

Firma

1. Per ogni $i = 1, \ldots, n$:
   a) Genera un valore casuale $\alpha_i \in \mathbb{Z}_q$
   b) Calcola $c_{i,\pi_i} = \mathcal{H}_n\left(\mathfrak{m}||\alpha_i G||i||\pi_i\right)$
   c) Per $j = \pi_i + 1, \ldots, m-1$, genera numeri causali $r_{i,j} \in \mathbb{Z}_q$ e calcola
   $$c_{i,j+1} = \mathcal{H}_n\left(\mathfrak{m}||r_{i,j}G - c_{i,j}K_{i,j}||i||j\right)$$
2. Per $i = 1, \ldots, n$, genera numeri causali $r_{i,m} \in \mathbb{Z}_q$ e calcola
   $$c_1 = \mathcal{H}_n\left(r_{1,m}G - c_{1,m}K_{1,m}|| \ldots ||r_{n,m}G - c_{n,m}K_{i,m}\right)$$
3. Per $i = 1, \ldots, n$:
   a) Per $j = 1, \ldots, \pi_i - 1$, genera numeri casuali $r_{i,j} \in \mathbb{Z}_q$ e calcola
   $$c_{i,j+1} = \mathcal{H}_n\left(\mathfrak{m}||r_{i,j}G - c_{i,j}K_{i,j}||i||j\right)$$
   interpretando il riferimento a $c_{i,1}$ come c_1.
   b) Calcola $r_{i,\pi_i} = \alpha_i + k_{i,\pi_i}c_{i,\pi_i}$

La firma sarà
$$\sigma = (c_1, r_{1,1}, r_{1,2}, \ldots, r_{1,m}, \ldots, r_{n,m})$$

Verifica
Come prima, sia $\mathfrak{m}$ l'hash del messaggio da firmare e $\mathcal{R} = \{K_{i,j}\}$ l'insieme delle chiavi delle firme.
La verifica di una chiave data è eseguita come segue:

1. Per $i = 1, \ldots, n$ e $j = 1, \ldots, m$ calcola:
   $$R'_{i,j+1} = r_{i,j}G - c'_{i,j}K_{i,j}$$
   $$c'_{i,j+1} = \mathcal{H}_n\left(\mathfrak{m}||R'_{i,j+1}||i||j\right)$$
   Interpretiamo ogni $c'_{i,1}$ come c_1
2. Calcoliamo $c'_1 = \mathcal{H}_n\left(R'_{1,m}|| \ldots ||R'_{n,m}\right)$

La firma sarà valida se $c'_1 = c_1$.

Correttezza

1. Per $j \neq \pi_i$ e per ogni i possiamo facilmente vedere che $c'_{i,j+1} = c_{i,j+1}$
2. Quando $j = \pi_i$, per ogni i

$$\begin{aligned} R'_{i,j+1} &= r_{i,j}G - c'_{i,j}K_{i,j} \\ &= \left(\alpha_i + k_{i,\pi_i}c'_{i,\pi_i}\right)G - c'_{i,\pi_i}K_{i,\pi_i} \\ &= \alpha_i G + k_{i,\pi_i}c'_{i,\pi_i}G - c'_{i,\pi_i}k_{i,\pi_i}G \\ &= \alpha_i G \end{aligned}$$

   In altre parole, $c'_{i,\pi_i+1} = \mathcal{H}_n\left(\mathfrak{m}||\alpha_i G||i||\pi_i\right) = c_{i,\pi_i+1}$

Possiamo così concludere che il passaggio della verifica identifica correttamente le firme valide.

Capitolo 4

Pedersen commitments

Come già abbiamo accennato, la privacy per Monero è importante anche per quanto riguarda l'ammontare delle transazioni: a questo scopo vengono utilizzati i **Pedersen commitments**. Il Pedersen commitment è un commitment che ha la caratterstica di essere additivo: in altre parole, se indichiamo con $C(a)$ e $C(b)$ i commitments per il valore a e il valore b rispettivamente, allora

$$C(a+b) = C(a) + C(b)$$

è il commitment per $a+b$.
Questa proprietà risulta essere fondamentale quando si scommette sugli amounts di transazioni, poiché si deve provare che gli input sono uguali agli output senza però rivelare il quantitativo di soldi. Inoltre i Pedersen commitments sono facili da implementare sulle curve ellittiche in quanto vale:

$$aG + bG = (a+b)G.$$

Definendo in un modo semplice come $C(a) = aG$, possiamo riconoscere immediatamente il commitment a 0. Per garantire la privacy è necessario aggiungere un *secret blinding factor* e un altro generatore H tali che non si conosca per quale valore di γ valga $H = \gamma G$ e, per la difficoltà del logaritmo discreto, sia impossibile calcolarla. Dunque, per evitare che sia facilmente riconoscibile il commitment a 0, possiamo definire il commitment di un amount a come

$$C(x, a) = xG + aH$$

dove x è il *blinding factor* che impedisce di indovinare a.
Nel caso di Monero come funzione hash viene usata $H =$ `to_point(Keddak(G))`, dove `Keddak` è un hashing algorithm mentre `to_point` è una funzione che mappa scalari in punti della curva.

4.1 Monero commitments

Una transazione in una cryptocurrency è una collezione di input e output che devono bilanciarsi. Per esempio, se Alice spende 100 unità di una certa moneta

(inputs), allora colui (o coloro) che riceve dovrebbe ricevere esattamente 100 unità. Nel caso in cui il pagamento a Bob fosse soltanto di 50 unità, allora un secondo output di 50 unità deve tornare ad Alice. Se abbiamo una transazione con n inputs $a_1, .., a_n$ e p outputs $b_1, .., b_p$, un osservatore esterno si aspetterebbe:

$$\sum_i a_i - \sum_j b_j = 0$$

Per la proprietà di additività dei commitment si ha:

$$\sum_i C_{i,in} - \sum_j C_{j,out} = 0$$

Questo servirà alla rete per verificare che colui che manda i soldi non stia spendendo più soldi di quanto abbia ricevuto in passato.
L'amount che deve essere speso corrisponde agli output della transazione precedente che ha come commitment:

$$C_j^a = x_i G + a_j H$$

Il mittente calcola dei nuovi commitments, chiamati *pseudo output commitments*, utilizzando lo stesso amount, ma cambiando i blinding factors:

$$C_j'^a = x_i' G + a_j H$$

Così facendo possiamo calcolare le chiavi private z_j attraverso la differenza dei due commitments:

$$C_j^a - C_j'^a = (x_j - x_j')G = z_j G$$

La precedente espressione si può riscrivere come $C_j^a - C_j'^a = z_j G + 0H$, dunque il mittente, che conosce le chiavi private z_j , è in grado di provare che nella somma non ci sia nessuna componente H da sommare. Inoltre il mittente calcola anche i commitment relativi agli output $\{b_t\}$:

$$C_t^b = y_t G + b_t H$$

Successivamente si devono scegliere i blinding factors tali che valga:

$$\sum_j x_j' - \sum_t y_t = 0$$

Se vale questa relazione allora è facile verificare [1] che:

$$\sum_j C_j'^a - \sum_b C_t^b = 0$$

[1] Sfruttando il fatto che $\sum_j a_j - \sum_t b_t = 0$, cioè che la somma degli inputs è uguale a quella degli outputs.

La scelta in Monero dei blinding factors è molto semplice, infatti si selezionano random tutti i fattori, tranne l'm-esimo pseudo output commitments che è calcolato come:

$$x'_m = \sum_t y_t - \sum_{j=1}^{m-1} x'_j$$

Per evitare che venga identificato colui che manda i soldi, Shen Noether [21] propose di verificare che i commitments sommassero ad un certo valore non nullo (zG).

$$\sum_i C_{i,in} - \sum_j C_{j,out} = zG$$
$$\sum_i (x_i G + a_i H) - \sum_j (y_j G + b_j H) = zG$$
$$\sum_i x_i - \sum_j y_j = z$$

Anche se la somma non sarà esattamente 0, ci riferiremo ad un commitment per una transazione valida con il termine *Commitments to zero*, ogni volta che la chiave privata z è conosciuta da colui che fa il commitment.

4.2 Range proofs

Un problema legato ai commitments additivi è che se noi abbiamo commitments per a, b, z e vogliamo usarli per provare $a + b = z$, allora questi saranno validi anche se sostituiamo ogni valore con l'inverso additivo modulo q, e questo potrebbe portare ad una creazione artificiale di soldi. La soluzione usata da Monero prima della versione v8 è di firmare i range di numeri usando la Borromean signature (riportata nel Capitolo 3 relativo alle Ring Signatures) nel modo seguente:

Per ogni amount a viene usata la sua rappresentazione binaria $a_0, a_1, \ldots, a_k$ tale che $a = a_0 2^0 + a_1 2^1 + \cdots + a_k 2^k$. In seguito vengono generati dei numeri casuali $x_1, \ldots, x_k \in \mathbb{Z}_q$ che svolgeranno il ruolo di blinding factors. Dunque vengono definiti i Pedersen commitments per ogni a_i : $C_i = x_i G + a_i 2^i H$ e se ne deriva il set di chiavi pubbliche $\{C_i, C_i - 2^i H\}$. Ovviamente una di queste chiavi pubbliche sarà uguale a $x_i G$, infatti:

- se $a_i = 0$ allora $C_i = x_i G + 0H = x_i G$
- se $a_i = 1$ allora $C_i - 2^i H = x_i G + 2^i H - 2^i H = x_i G$

Dunque un blinding factor x_i sarà sempre la chiave privata corrispondente a uno tra $\{C_i, C_i - 2^i H\}$. É quindi possibile firmare un amount a in una transazione usando la Borromean Ring Signature riportata nel capitolo precedente con il ring:

$$\{\{C_0, C_0 - 2^0 H\}, \ldots, \{C_k, C_k - 2^k H\}\}$$

La soluzione adottata in seguito da Monero per provare che ogni amount sta in un certo range (da 0 a $2^{64} - 1$) usando i Bulletproofs, descritta da Benedikt Bünz et al. in [6, 11, 28], esula dallo scopo di questo elaborato e per questo non verrà trattata.

Capitolo 5

Address

Ogni nodo della blockchain di Monero è identificato da una coppia di chiavi private (k^v, k^s), con $k^v, k^s \in \mathbb{Z}_q$, e dalle corrispondenti chiavi pubbliche (K^v, K^s), le quali saranno due punti della curva. In particolare definiamo:

- *address*: la coppia di chiavi pubbliche (K^v, K^s);
- *view key*: la chiave privata k^v, la quale permette ad un utente di vedere se il suo address è output di una qualche transazione nella blockchain, garantendo il possesso di una quantità di criptomoneta;
- *spend key*: la chiave privata k^s, che permette ad un utente di poter spendere un determinato output e controllare che esso non sia ancora stato speso, impedendo il double-spending.

Nella maggioranza dei casi la view key è calcolata a partire dalla spend key nel seguente modo $k^v = \mathcal{H}_n(k^s)$, dove $\mathcal{H}_n$ è una funzione hash, così da dover memorizzare solo una delle due chiavi. Invece, k^s è generata da una serie di 25 parole casuali (dove la 25° parola è un checksum).

5.1 One-time address

Una particolarità delle transazioni di Monero è che l'address del destinatario non compare mai direttamente, ma è utilizzato uno scambio di informazioni Diffie-Hellman tra i due nodi coinvolti per generare un *one-time address* unico per ogni output di una transazione. É stato introdotto questo sistema per impedire ai nodi esterni alla transazione di capire a quale address sia destinato un particolare output.

Vediamo come costruirlo nel caso semplice di una transazione in cui Alice vuole mandare 1 XMR a Bob. Alice conosce l'address di Bob (K_B^v, K_B^s), al quale è associata la coppia (k_B^v, k_B^s). Il protocollo da seguire è (vedi [25]):

1. Alice genera un numero casuale $r \in \mathbb{Z}_q$ e calcola il *one-time* address:
$$K^o = \mathcal{H}_n(rK_B^v)G + K_B^s$$
Il punto rG è chiamato *transaction public key* ed è unico per ogni transazione.
2. Alice imposta K^o come address di destinazione dell'output e aggrega la transaction public key rG ai dati della transazione. Il tutto lo inserisce nella rete.
3. Bob riceve i dati della transazione di Alice e usa rG e la sua *view key* k_B^v per calcolare $K_B'^s = K^o - \mathcal{H}_n(rk_B^vG)$. Successivamente Bob verifica che $K_B'^s = K_B^s$ e se la verifica è andata a buon fine allora sa di essere lui il destinatario della transazione.
4. Bob calcola la *one-time key*:
$$k^o = \mathcal{H}_n(rk_B^vG) + k_B^s$$

Si può osservare che solo chi conosce la *view key* è in grado di vedere il reale destinatario della transazione. Mentre la *one-time key* calcolata serve a Bob quando decide di voler spendere gli XMR ricevuti da Alice. Infatti osserviamo che nell'espressione di k^o compare la *spend key* di Bob.

Senza k^o Alice non può calcolare l'immagine della chiave di output, quindi non può mai sapere con certezza se Bob spende gli XMR che gli ha inviato.[1]

5.2 Transazione multioutput

Consideriamo il caso in cui Alice deve fare una transazione che comprende p output. Siano $(K_{B_i}^v, K_{B_i}^s)$ gli address degli utenti B_i, con $i \in \{1, 2, \ldots, p\}$. Il protocollo che deve eseguire Alice è formalmente lo stesso del caso con un singolo output[2], l'unico passaggio diverso è la generazione dei *one-time address*, infatti, affinché ogni output sia unico, Alice deve calcolare:
$$K_i^o = \mathcal{H}_n(rK_{B_i}^v||i)G + K_{B_i}^s \quad \forall i \in \{1, 2, ..., p\}$$

[1] Immagina che Alice produca due transazioni, ciascuna contenente lo stesso one-time address di output K^o che Bob può spendere. Poiché K^o dipende solo da r e K_B^v, non c'è motivo per cui non possa farlo. Bob può spendere solo uno di questi output perché ogni indirizzo occasionale ha solo un'*image key*, quindi se non sta attento Alice potrebbe ingannarlo. Potrebbe effettuare la transazione 1 con molti soldi per Bob e successivamente la transazione 2 con una piccola somma per Bob. Se spendesse i soldi in 2, non potrà mai spendere i soldi in 1. Infatti, nessuno potrebbe spendere i soldi in 1, "bruciandoli" di fatto. I portafogli Monero sono stati progettati per ignorare l'importo minore in questo scenario.

[2] In realtà, a partire dal protocollo v12 sono richiesti due output per ciascuna transazione (non-miner), anche se significa che un output ha un importo 0. Ciò migliora l'indistinguibilità delle transazioni mescolando i casi a 1 uscita con le transazioni a 2 uscite molto più comuni. L'implementazione principale invia output di importo 0 a un indirizzo casuale.
Dal protocollo v8 il numero massimo di output è limitato a 16.

Dunque gli output sono indicizzati e successivamente l'indice i dell'output $i-esimo$ è inserito come argomento della funzione hash insieme al punto $rK^v_{B_i}$. Si può osservare che la transaction public key rG è unica per tutti gli output. Inoltre Alice quando deve inviare la transazione alla rete tra i dati da aggregare oltre a rG deve anche inviare l'indice i associato al corrispondente one-time address K^o_i.

5.3 Subaddress

Gli utenti di Monero hanno anche la possibilità di generare dei subaddresses dal proprio address di partenza. Lo scopo è dare la possibilità ad ogni nodo di gestire i fondi anche in modo differenziato, ciò può essere utile per non collegare tutte le proprie attività all'interno della rete. Un'altra applicazione di un subaddress può essere quella di creare un fondo che serve per pagare spese particolari. Le caratteristiche principali di un subaddress sono:

- ad un utente basta conoscere solo la propria coppia (k^v, k^s) per accedere a tutti i suoi subaddress.
- la difficoltà a risalire all'address corrispondente dal subaddress da parte di un nodo esterno è la stessa per risolvere il problema DLP.

Dunque un utente avente un address (K^v, K^s) può derivare il suo i-esimo subaddress calcolando:

$$K^{s,i} = K^s + \mathcal{H}_n(k^v||i)G$$
$$K^{v,i} = k^v K^{s,i}$$

Se Alice vuole mandare un pagamento di '0' ad un subaddress di Bob $(K^{v,1}_B, K^{s,1}_B)$ i passaggi sono i seguenti:

1. Alice genera un numero cauale $r \in \mathbb{Z}_q$ e calcola il *one-time address*:

$$K^o = \mathcal{H}_n(rK^{v,1}_B||0)G + K^{s,1}_B$$

2. Alice imposta K^o come address di destinazione dell'output, aggrega $rK^{s,1}_B$ e l'amount '0' ai dati della transazione. Il tutto lo inserisce nella rete.

3. Bob riceve i dati della transazione di Alice e usa $rK^{s,1}_B$, la sua *view key* k^v_B e il valore '0' per calcolare $K'^s_B = K^o - \mathcal{H}_n(rK^{v,1}_B||0)$. Successivamente Bob verifica che $K'^s_B = K^{s,1}_B$ e se la verifica è andata a buon fine allora sa che il pagamento è destinato al subaddress 1.

4. Bob calcola la *one-time key*:

$$k^o = \mathcal{H}_n(rK^{v,1}_B||0) + k^{s,1}_B$$

dove $k^{s,1}_B = k^s_B + \mathcal{H}_n(k^v_B||0)$.

Osserviamo che in questo caso Alice ha generato una transaction public key $rK_B^{s,1}$, che è specifica del subaddress di Bob. Dunque se Alice dovesse mandare p-output e tra i destinatari ci fosse almeno un subaddress, dovrebbe generare p transaction public keys. Ad esempio se Alice dovesse trasferire XMR ad un subaddress di Bob $(K_B^{v,1}, K_B^{s,1})$ e all'address di Carl (K_C^v, K_C^s), allora dovrebbe generare due numeri casuali r_1 e r_2 e aggiungere le due transaction public keys $\{r_1 K_B^{s,1}, r_2 G\}$ ai dati della transazione.

Capitolo 6

Ring confidential transactions

In questo Capitolo descriviamo come sono strutturate le transazioni nella corrente versione di Monero. Ogni transazione segue uno schema *Ring Confidential Transactions*, o *RingCT*. In particolare Monero adotta la versione *RCTTypeBulletProof2*, implementata nel protocollo (v10), la quale è un miglioramento della *RCTTypeSimple*. L'aggiornamento introduce un nuovo meccanismo di controllo *BulletProof*, il quale è un short non-interactive zero-knowledge proofs. Questo cambiamento è stato fatto in quanto la verifica BulletProof alleggerisce notevolmente il peso di una singola transazione, e di conseguenza anche della relativa fee, rispetto alla *Range Proof* che era adottata in precedenza.

Ipotizziamo di aver ricevuto in passato degli amounts che diventano i nostri inputs $a_1, ..., a_m$ associati ai rispettivi one-time addresses $K^o_{\pi,1}, ..., K^o_{\pi,m}$ e ai commitments $C^a_{\pi,1}, ..., C^a_{\pi,m}$. Questi sono nella forma:

$$C^a_{\pi,j} = x_j G + a_j H \quad j \in \{1, ..., m\}$$

dove $\{x_j\}$ sono i blinding factors delle transazioni precedenti.

Dato che è necessario incentivare con delle fees i miners a inserire le transazioni all'interno di un blocco, solitamente la somma totale degli outputs che vogliamo inviare è sempre minore della somma degli inputs. In particolare definiamo f il valore della nostra fee. Quest'ultimo deve essere inserito in chiaro tra i dati della transazione. Dunque, se usiamo come inputs $a_1, ..., a_m$ e gli outputs sono $b_0, ..., b_{p-1}$, deve valere la relazione:

$$\sum_{j=1}^{m} a_j - \sum_{t=0}^{p-1} b_t - f = 0 \tag{6.1}$$

Ora calcoliamo gli pseudo output commitment $C'^{a}_{\pi,1}, ..., C'^{a}_{\pi,m}$:

$$C'^{a}_{\pi,j} = x'_j G + a_j H \quad j \in \{1, ..., m\}$$

e salviamo i valori $(z_1 = x_1 - x'_1, ..., z_m = x_m - x'_m,)$. Successivamente generiamo i nuovi commitment $C^b_0, ..., C^b_{p-1}$ da associare agli output.
Noi usiamo le chiavi private $z_1, ..., z_m$ per verificare il commitment a zero $(C^a_{\pi,1} - C'^{a}_{\pi,1}), ..., (C^a_{\pi,m} - C'^{a}_{\pi,m})$. Dato che deve continuare a valere la relazione (6.1) calcoliamo anche il commitment della fee $C(f) = fH$, senza usare *blinding factors*, e otteniamo:

$$(\sum_{j=1}^{m} C'^{a}_j - \sum_{t=0}^{p-1} C^b_t) - fH = 0 \tag{6.2}$$

Successivamente selezioniamo m insiemi di cardinalità v,[1] composti da one-time addresses e i rispettivi commitment presi casualmente dalla blockchain, i quali sono utilizzati come finti input per mascherare quello vero. Dalla versione v12 è previsto che questi siano presi da blocchi distanti almeno 10 dall'ultimo inserito. Si costruiscono m rings nella forma:

$$\begin{aligned} R_j = \{&\{K^o_{1,j}, (C_{1,j} - C'^{a}_{\pi,j})\}, \\ &... \\ &\{K^o_{\pi,j}, (C^a_{\pi,j} - C'^{a}_{\pi,j} = z_j G)\}, \\ &... \\ &\{K^o_{v+1,j}, (C_{v+1,j} - C'^{a}_{\pi,j})\}\} \end{aligned}$$

dove π è l'indice che indica il vero input della transazione. In questo modo possiamo firmare il ring attraverso il protocollo MLSAG dove vogliamo utilizzare come chiavi private la coppia $(k^o_{\pi,j}, z_j)$. Di conseguenza la firma per l'input *j-esimo* sarà nella forma:

$$\sigma_j(\mathfrak{m}) = (c_1, r_{1,1}, r_{1,2}, ..., r_{v+1,1}, r_{v+1,2})$$

a cui si associa la key image $\widetilde{K}^o_j$.
Osserviamo che per la chiave privata z_j non si calcola la key image corrispondente, in quanto non è necessaria per la verifica del commitment a zero.
La decisione di firmare ogni input separatamente permette di non avere l'indice π uguale per tutti e nel caso in cui qualcuno dovesse venire a conoscenza della posizione di un vero input all'interno di uno dei ring R_j non ha comunque informazioni sulle posizioni reali dei rimanenti.
Il messaggio $\mathfrak{m}$ da firmare per ogni input è l'hash corrispondente ai dati di tutta la transazione, eccetto le firme MLSAG che devono ancora essere calcolate. Possiamo concludere che è prodotto un unico messaggio $\mathfrak{m}$ e si calcolano m firme.
É importante notare che quando si firma il messaggio si utilizzano le one-time keys, le quali tengono traccia della nostra spend key, che prova che siamo noi

[1] Adesso v è pari a 10 in modo tale da standardizzare le transazioni

gli effettivi proprietari degli inputs.
La rete inoltre può verificare facilmente se gli inputs non siano già stati spesi. Questo è possibile grazie al fatto che nella firma si calcola la key image, la quale è unica per ogni one-time address. Quindi anche se avessimo due firme differenti su uno stesso input, la key image prodotta sarebbe la stessa, in quanto è calcolata in modo deterministico dalla coppia (k^o, K^o) e non dipende dagli altri elementi del ring o dagli altri inputs. Dunque è sufficiente verificare che nella blockchain non sia già presente la stessa key image in un'altra transazione.

Riassumiamo la struttura di una generica transazione nel seguente schema:

- *Type*: questa voce specifica il tipo di transazione e può assumere due valori '0' o '4'. '0' si riferisce alla transazione che genera il reward del miner che attacca il blocco, mentre '4' è per le transazioni tra gli utenti.
- *Inputs*: per ogni input $j \in \{1, \dots, m\}$ si registra
  - *Ring member offsets*: lista di informazioni che indicano dove un utente della rete può trovare i dati del membro *i-esimo* del ring all'interno della blockchain, così da poter verificare le informazioni
  - *MLSAG signature*: la firma σ_j e i termini c_1, $r_{i,1}$ e $r_{i,2}$ per ogni $i \in \{1, \dots, v+1\}$
  - *Key image*: $K_j^{o,a}$
  - *Pseudo output commitment*: $C_j'^a$
- *Outputs*: per ogni output $t \in \{0, \dots, p-1\}$
  - *One-time address*: $K_t^{0,b}$
  - *Output commitment*: C_t^b
  - L'amount cifrato
  - *BulletProof*: verifica che l'output sia accettabile
- *Transaction fee*: comunicata in un *cleartext* e deve essere inserita in unità atomiche di XMR, dunque il valore moltiplicato per 10^{12}
- Dati extra della transazione: *transaction public key*

Capitolo 7

Kovri

Finora abbiamo illustrato come Monero nasconde le informazioni relative alle transazioni: le *ring signatures* proteggono l'identità del mittente, mentre quella del destinatario è protetta grazie all'uso degli *stealth address*; inoltre, l'ammontare di ogni transazione è protetto dalle *ring confidential transactions*.
Monero [2] è una rete composta di nodi che comunicano tra loro scambiandosi messaggi (transazioni) da indirizzi IP. L'indirizzo IP è come se fosse l'abitazione virtuale di ciascun utente, il che è un dato a dir poco sensibile e se venisse scoperto da qualche hacker malevolo potrebbe portare conseguenze gravi per l'utente in questione.
Purtroppo le accortezze usate in Monero per proteggere la privacy dei propri utenti non bastano per garantire la completa sicurezza, perchè quando una persona invia una transazione ha l'IP esposto in rete; si potrebbe provare a nasconderlo utilizzando una VPN o TOR, ma non è abbastanza, perchè VPN è suscettibile ad attacchi basati su correlazione mentre TOR possiede delle autorità centrali fidate che vanno quindi contro il concetto fondamentale di rete decentralizzata.

É qui che entra in gioco **Kovri**. Kovri è una tecnologia di anonimizzazione gratuita [2], non è basata su TOR ma sulle specifiche di I2p *Invisible Internet Project*, un software libero e Open Source per la realizzazione di una rete anonima. Utilizza teniche sofisticate di cifratura e di routing per creare una sovrastruttura di rete privata su internet; questa sovrastruttura protetta permette agli utenti di nascondere la propria identità e il proprio indirizzo IP.

In pratica, Kovri convoglia il traffico attraverso la rete I2p utilizzando una cosiddetta "cifratura ad aglio"; quando un utente vuole inserire nella rete di Monero una transazione, la cifra e la inoltra agli altri nodi. Le informazioni sul suo indirizzo IP viaggiano con le transazioni sulla sovrastruttura di rete citata in precedenza, i quali vengono cifrati in più strati ogni volta che giungono ad un diverso nodo della rete, più o meno come il funzionamento di una matrioska: ogni bambola interna possiede un lucchetto che rende impossibile per un esterno

Figura 7.1: *Schema matrioska* [20]

leggerne il contenuto. In questo modo i nodi della rete non hanno la possibilità di leggere il messaggio che viene inoltrato, poiché possono leggere soltanto le istruzioni per inoltrare il messaggio al nodo successivo.

In futuro Kovri verrà reso disponibile nei nuovi rilasci di Monero, eventualmente implementando delle API che ne permettano l'uso ad altre criptomonete. Molti ritengono che possa rivoluzionare in futuro le connessioni peer-to-peer, aumentando notevolmente la resilienza e la privacy della rete, in modo che eventuali nodi malevoli non possano più minacciare la sicurezza degli utenti né bloccare la trasmissione delle transazioni sulla rete.

Bibliografia

[1] Kurt M. Alonso and Jordi Herrera Joancomartí. *Monero - Privacy in the Blockchain*. Universitat Autónoma de Barcelona, May 2018. https://eprint.iacr.org/2018/535.

[2] Filippo Angeloni. *Che cos'è Monero: la guida definitiva semplice alla portata di tutti*, 2021. https://filippoangeloni.com/che-cose-monero-la-guida-definitiva-semplice-alla-portata-di-tutti/.

[3] Daniel J. Bernstein and Tanja Lange. *Faster Addition and Doubling on Elliptic Curves*. Springer Berlin Heidelberg, 2007.

[4] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. *Twisted Edwards Curves*. Springer Berlin Heidelberg, 2008.

[5] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. *High-speed high-security signatures*. Journal of Cryptographic Engineering, September 2012.

[6] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. *Bulletproofs: Short Proofs for Confidential Transactions and More*. Cryptology ePrint Archive, Report 2017/1066, 2017. https://eprint.iacr.org/2017/1066.

[7] Francisco Cabañas. *Francisco Cabanas - Critical Role of Min Block Reward Trail Emission - DEF CON 27 Monero Village*, December 2019. https://www.youtube.com/watch?v=IlghysBBuyU.

[8] Francisco Cabañas. *Lightning talk: An Overview of Monero's Adaptive Blockweight Approach to Scaling*, December 2019. https://frab.riat.at/en/36C3/public/events/125.html.

[9] Francisco Cabañas. *MoneroKon 2019 - Spam Mitigation and Size Control in Permissionless Blockchains*, June 2019. https://www.youtube.com/watch?v=Hbm0ub3qWw4.

[10] CoinMarketCap. *Monero*, June 2021. https://coinmarketcap.com/it/currencies/monero/.

[11] dalek cryptography. *Bulletproofs*, 2021. https://doc-internal.dalek.rs/bulletproofs/index.html.

[12] JollyMort. *Monero Dynamic Block Size and Dynamic Minimum Fee*, March 2017. https://github.com/JollyMort/monero-research/blob/master/Monero%20Dynamic%20Block%20Size%20and%20Dynamic%20Minimum%20Fee/Monero%20Dynamic%20Block%20Size%20and%20Dynamic%20Minimum%20Fee%20-%20DRAFT.md.

[13] Koe, Kurt M. Alonso, and Sarang Noether. *Zero to Monero: Second Edition*, April 2020. https://www.getmonero.org/library/Zero-to-Monero-2-0-0.pdf.

[14] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. *Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups*. Cryptology ePrint Archive, Report 2004/027, 2004. https://eprint.iacr.org/2004/027.

[15] Gregory Maxwell and Andrew Poelstra. *Borromean ring signatures **, 2015.

[16] Monero. *Monero cryptonight variants, and add one for v7*, April 2018. https://github.com/monero-project/monero/pull/3253.

[17] Monero. *Monero 0.13.0 "Beryllium Bullet" Release.*, 2018. https://www.getmonero.org/2018/10/11/monero-0.13.0-released.html.

[18] Monero. *Monero 0.14.0 "Boron Butterfly" Release.*, 2019. https://web.getmonero.org/2019/02/25/monero-0.14.0-released.html.

[19] Monero. *Randomx*, 2019. https://www.monerooutreach.org/stories/RandomX.php.

[20] Monero. *Monero: Kovri (come Monero nasconde gli indirizzi IP)*, 2021. https://www.youtube.com/watch?v=cxgbLI6IZGs&ab_channel=Monero.

[21] Shen Noether. *Ring signature confidential transactions for monero*, 2015. http://eprint.iacr.org/2015/1098.

[22] Shen Noether, Adam Mackenzie, and the Monero Research Lab. *Ring confidential transactions*, December 2016. https://ledger.pitt.edu/ojs/ledger/article/view/34.

[23] Monero Core Team. *Monero*, 2021. https://web.getmonero.org/it/.

[24] Reddit users. *Monero v0.9.3 - Hydrogen Helix - released!*, 2016. https://www.reddit.com/r/Monero/comments/4bgw4z/monero_v093_hydrogen_helix_released_urgent_and/.

[25] Nicolas van Saberhagen. *CryptoNote v2.0*, October 2013. https://bytecoin.org/old/whitepaper.pdf.

[26] Wikipedia. *Monero*, 2021. https://en.wikipedia.org/wiki/Monero.

[27] Howard "hyc" Chu. *RandomX, Pull Request #5549*, May 2019. `https://github.com/monero-project/monero/pull/5549`.

[28] Adam "waxwing" Gibson. *From Zero (Knowledge) To Bulletproofs*, March 2018. `https://github.com/AdamISZ/from0k2bp/blob/master/from0k2bp.pdf`.

FINANZA DECENTRALIZZATA

Kairi Zuccarino, Gianluca Mega, Tommaso Toso, Adriano Koleci

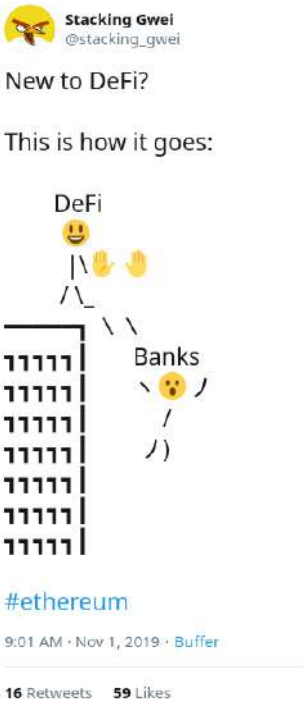
Introduzione alla Finanza Decentralizzata

Che cos'è la *DeFi*?

Il grande successo riscosso da Bitcoin, il conseguente boom delle altcoin e l'innovazione apportata dalla tecnologia blockchain rappresentano serie premesse per una ulteriore diffusione delle criptomonete e degli strumenti ad esse legati anche in altri settori dell'economia e della finanza. Molti di coloro che già hanno fatto il loro ingresso nel mondo della criptoeconomia sono estremamente affascinati da tale prospettiva ed è un mercato che vale già qualche decina di miliardi di dollari [1]. Ciò è dovuto al fatto che, in uno scenario di questo tipo, gli scambi ed il funzionamento degli strumenti finanziari non sono regolati da alcuna autorità centrale. I vantaggi principali sono due: il primo è una maggior libertà nella natura delle transazioni effettuate; il secondo è l'eliminazione dei prezzi di intermediazione. L'insieme delle soluzioni finora implementate che cercano di ricreare servizi finanziari consolidati senza la presenza di un intermediario o una autorità centrale sfruttando la tecnologia blockchain è detta finanza decentralizzata, più comunemente chiamata Decentralized Finance o, per gli amici, DeFi. Ad oggi, sono stati fatti tentativi e implementate soluzioni per cercare di replicare i seguenti strumenti finanziari:

- Exchange di asset;
- Depositi e prestiti;
- strumenti assicurativi;
- gestione di portafogli.

Prima di poter procedere alla decentralizzazione di queste pratiche, è necessario definire come verrà effettuata la gestione di alcuni aspetti fondamentali, come ad esempio i tassi di interesse e i casi di insolvenza. Tali problematiche sono piuttosto semplici da trattare quando esiste un'autorità centrale, poichè è essa ad imporre le condizioni per la realizzazione di una di queste pratiche: i prestiti, ad esempio, non sono concessi a tutti, ma vengono valutati attentamente in base a chi li richiede. In questo scenario, la sicurezza degli strumenti corrisponde, di fatto, alla sicurezza e all'affidabilità dell'autorità centrale stessa. Eventi anche recenti (si pensi alla questione Lehman Brothers nella crisi finanziaria del 2008) hanno palesato le criticità di un sistema così strutturato.



DeFi rispetto alla finanza tradizionale
[2]

Nella nostra tesina ci concentreremo sulle soluzioni volte a sostituire gli exchange di asset e sui depositi e prestiti collateralizzati nel mondo Crypto. Tali strumenti vengono offerti, rispettivamente, da due piattaforme di successo quali *Uniswap* e *AAVE*. Per prima cosa, procederemo a descrivere gli exchange ed i

prestiti collateralizzati.

CEX e DEX

I centri di scambio tradizionali riuniscono in un unico luogo compratori e venditori di stock e valute, assicurando ad essi di poter entrare ed uscire da una posizione in maniera relativamente facile, rifornendo di liquidità il mercato. Storicamente, gli exchange di asset e valute sono sempre stati di natura centralizzata, in quanto l'exchange stesso si poneva come intermediario per lo scambio degli strumenti finanziari e del denaro. I primi exchange per lo scambio di beni digitali si svilupparono anch'essi ponendosi come un'autorità centrale che si occupa di fare da intermediario fra le parti coinvolte in una transazione. Tale tipologia di exchange di beni digitali è denominata CEX (Centralized Exchange). Un aspetto di particolare interesse dei CEX è che gli utenti, nel momento in cui decidono di voler entrare nell'exchange, cedono la gestione dei propri fondi alla piattaforma (depositando i propri token in un wallet detenuto dall'exchange), rinunciando temporaneamente alla proprietà degli asset che vogliono scambiare (più precisamente delle chiavi private che permettono di usufruire di quel bene digitale). È immediato comprendere che uno dei problemi legati a questa pratica riguarda l'affidabilità (sia a livello di sicurezza informatica che di onestà) dell'exchange. Generalmente, questo è un rischio che la maggior parte degli utenti decide di accettare, soprattutto se si ha a che fare con piattaforme molto famose e tecnicamente avanzate. Gli exchange decentralizzati, detti DEX, al contrario, non ricorrono a organizzazioni intermedie per la realizzazione delle transazioni e prevedono un non-custodial framework in cui gli utenti mantengono la proprietà dei beni da scambiare. La decentralizzazione dell'exchange fa diventare lo stesso un mercato virtuale dove venditori e acquirenti si incontrano e le transazioni sono regolate da smart contracts. A differenza dei CEX, dunque, gli utenti continuano ad essere proprietari dei fondi. [3] Riportiamo ora un breve elenco dei vantaggi e svantaggi di ricorrere ai DEX.

- **Vantaggi dei DEX**
  - *non-custodial framework*: un DEX prevede l'utilizzo di smart contract per la realizzazione degli scambi, senza richiedere la cessione temporanea dei beni;

  - *varietà degli asset*: molte altcoin sono accessibili solo attraverso dei DEX, dove le transazioni peer-to-peer possono avvenire anche in assenza di grandi volumi di scambio nell'exchange. Di fatto, questo aumenta il livello di inclusione finanziaria;
  - *sicurezza*: la maggior parte dei DEX esegue e memorizza le transazioni sulla blockchain su cui gli smart contract vengono eseguiti, di fatto riducendo la probabilità di attacchi informatici;
  - *costi minori*: essendo basati su smart contracts eseguiti in modo automatico, i DEX richiedono generalmente delle fee molto più basse dei CEX centralizzati;
  - *privacy*: le chiavi private non vengono passate ad un garante, di fatto rimangono a conoscenza del solo proprietario.

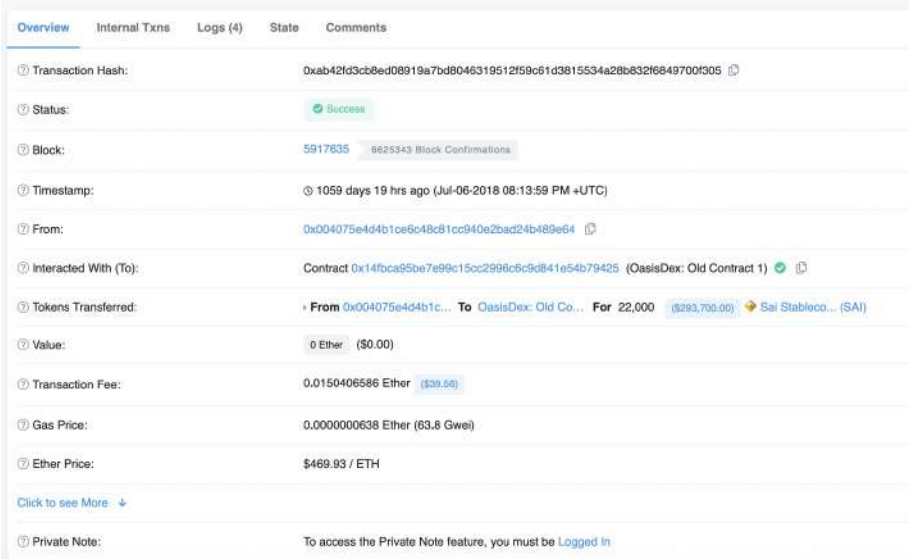
- **Svantaggi dei DEX**
  - *scarsa scalabilità*: i DEX impiegano smart contracts che vivono su una certa blockchain (in molti casi è quella di Ethereum), dunque la loro velocità di processazione dipende da quest'ultima (per Ethereum 15 TPS i.e. Transactions per Second);
  - *bassa usabilità*: è richiesto un buon livello di competenze informatiche per poter interagire con un DEX (uso di piattaforme wallet esterne, implementazione di un collegamento fra quest'ultime e l'interfaccia del DEX). Nel caso dei CEX, invece, le piattaforme facilitano ogni aspetto del trading;
  - *liquidità*: non essendo ancora utilizzati in larga misura, la liquidità di questi exchange è piuttosto bassa, dunque può non essere sempre possibile trovare una controparte con cui effettuare transazioni.

Dopo aver evidenziato le principali innovazioni apportate dall'approccio decentralizzato, si passa ora ad illustrare le diverse tipologie di DEX.

DEX con order Book *on chain*

L'order book è il registro contenete tutti gli ordini di acquisto e vendita di un titolo correntemente piazzati. Il metodo sicuramente più trasparente per procedere con le transazioni è un sistema di order book *on chain*: in questo

caso, il registro degli ordini è collocato sulla blockchain. Gli ordini sono visibili dunque a tutti gli utenti. Nel momento in cui c'è corrispondenza fra ordini di acquisto e di vendita, uno smart contract esegue la transazione, scambiando gli asset fra le due o più parti coinvolte. Si può osservare come tale approccio sia altamente decentralizzato. Tuttavia, esso comporta anche delle limitazioni significative. Infatti, per ogni movimento è necessario operare sulla blockchain e quindi ogni volta si devono pagare delle commissioni per registrare la transazione (commissioni che possono essere piuttosto alte, come questa transazione:



esempio di transazione su Oasis DEX

Inoltre bisogna aspettare i tempi di inserimento del blocco, il che rende tutto il processo molto meno rapido e sicuramente poco scalabile.

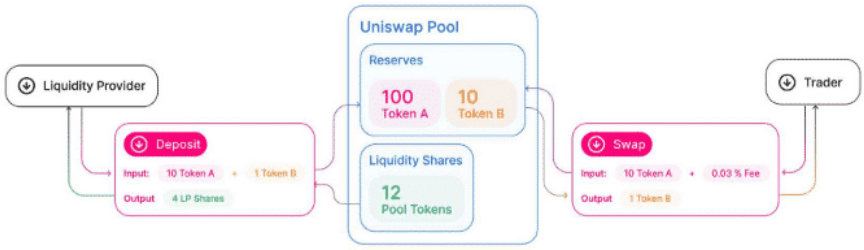
DEX con order Book *off-chain*

L'approccio degli order book *off chain* rinuncia ad una parte del livello di decentralizzazione e trasparenza raggiunto dalla soluzione precedente per aumentare la scalabilità del processo. In questo caso, la gestione dell'order book è affidata ad un'autorità centrale (solitamente il gestore dell'exchange) che si occupa di abbinare le offerte di acquisto e quelle di vendita. La transazione viene però eseguita on-chain tramite l'utilizzo di smart contract ed altri strumenti. Gli

order book *off chain* possono essere centralizzati (come IDEX o Switcheo) oppure distribuiti, come gli smart contract sul protocollo 0x per token ERC-20 che operano sulla catena una volta che le due parti sono state accoppiate.

Automated Market Maker

Una soluzione che cerca di superare gli order book ed i problemi ad essi legati sono gli **Automated Market Maker** o **AMM**[4]. che cambia radicalmente l'approccio rispetto agli order book: viene creato uno smart contract chiamato *liquidity pool* che blocca dei fondi che poi verranno scambiati. Gli AMM sono di fatto costitutiti da un insieme di smart contract, detti *liquidity pool*, contenenti un certo ammontare di due criptovalute.



Esempio di liquidity pool di Uniswap [5]

Per effettuare gli scambi, gli utenti interagiscono direttamente con questi smart contract, depositando una certa quantità dell'asset che vogliono vendere e ritirando il corrispettivo dell'asset che bogliono acquistare. Gli utenti non interagiscono più con altre persone dunque, bensì effettuano transazioni con uno smart contract. Una liquidity pool è uno smart contract contenente una riserva di criptovalute o token crowdsourced con il quale è possibile interagire e scambiare beni digitali. L'idea da cui nascono è quella di sopperire al problema di liquidità di un exchange. Gli exchange che ne fanno uso incentivano il sourcing della pool proponendo degli incentivi a coloro che la alimentano con i token necessari, detti liquidity provider. Tale aspetto è fondamentale, poiché maggiore è la liquidità in un pool, più facili sono gli scambi sull'exchange. Il modello più semplice di liquidity pool è certamente quello con due token e, a differenza di prima, il prezzo dei beni all'interno della pool non è collegato direttamente a

quello del mercato. Esso viene regolato dalla liquidity pool stessa secondo una certa formula, ad esempio:

$$balance_{tokenA} * balance_{tokenB} = k$$

E' chiaro che il prezzo di ogni bene varia in base alla sua disponibilità nel pool. Chi decide di depositare liquidità riceve un compenso per il deposito (un tasso di interesse) proporzionale al contributo nella liquidity pool, mentre chi decide di scambiare dovrà pagare una commissione che verrà distribuita tra tutti i liquidity providers. La non corrispondenza tra i prezzi della liquidity pool e quelli del mercato può dare luogo alla pratica dell'arbitraggio, con il rischio per i liquidity provider di rimettere del denaro. Supponiamo che Alice sia il liquidity provider di un pool che gestisce ETH e BTC e supponiamo che il prezzo dei BTC crolli all'infuori del pool (ricordiamo che il prezzo nel pool non è deciso dal mercato ma dalla quantità presente). Tale cambio nel prezzo dei BTC non si trasferisce automaticamente all'interno della pool, poiché il suo prezzo è stabilito solo dalla quantità di BTC presenti al suo interno. Bob a questo punto può acquistare BTC nel mercato, scambiare ad un prezzo vantaggioso BTC con ETH e rivendere ETH per BTC nella pool, così da ottenere un guadagno netto. Questo sembra minacciare gravemente il sistema delle liquidity pool, in quanto i providers avranno poco interesse a mettere a disposizione la propria liquidità. Tale pratica è ampiamente diffusa, tanto che sono presenti degli strumenti per automatizzare il processo (https://hummingbot.io/blog/2020-12-amm-arbitrage-uniswap-balancer/). Generalmente, si cerca di contrastare la pratica dell'arbitraggio imponendo delle fee molto alte in caso di estrema variabilità del mercato, ma può anche essere funzionale, come nel caso di Uniswap.

Metodo	Pro	Contro
Order Book On Chain	Trasparenza Decentralizzazione	Lentezza
Order Book Off Chain	Velocità	No trasparenza Fiducia in terzi
AMM	Velocità Trasparenza Decentralizzazione	Arbitraggio

Prestiti

Il secondo punto che andremo ad analizzare riguarda i meccanismi atti ad offrire prestiti tramite la blockchain. Per parlare di questo è necessario dapprima introdurre il *peer-to-peer lending*. Il peer-to-peer lending consiste nella concessione di prestiti senza intermediazione di terze parti (ad esempio una banca), a partire da un sistema di domanda e offerta degli utenti. Solitamente, tale pratica viene anche indicata con il nome di *crowdlending*. Negli anni sono stati numerosi i servizi che hanno reso accessibile questa tipologia di prestiti, diffondendola a tal punto che si prospetta che entro il 2023 questo mercato varrà quasi 400 miliardi di dollari, con un incremento del 75% rispetto a pochi anni fa [6]. Ciò è dovuto anche a causa dei requisiti più stringenti che le banche impongono ai privati o alle piccole e medie imprese per ottenere i prestiti. Come si può vedere, il P2P lending è già in un certo senso decentralizzato e i servizi attualmente a disposizione si occupano di accordare le due parti per effettuare lo scambio. Sulla blockchain, sempre grazie agli smart contracts, è possibile ricostruire completamente il meccanismo di concessione di prestiti sotto forma di criptovalute, in maniera più semplice ed automatizzata.
Un primo modo, dunque, per offrire prestiti con le criptovalute, è quello del P2P lending, dove gli utenti mettono a disposizione i propri fondi per prestiti collateralizzati, dove chi contrae il debito mette una valuta fiat o una criptovaluta come garanzia (non dissimile da un prestito con ipoteca o una carta di credito di tipo *secured*). I prestiti, dunque, non potranno mai superare la liquidità messa a garanzia. Coloro che mettono a disposizione le proprie criptomonete, i *P2P lenders*, ricevono degli interessi da chi ha contratto il prestito per il sevizio svolto. Ciò che non bisogna dimenticare è che in questo caso i prestiti non vengono erogati conoscendo il cliente e cercando di prevedere se sarà insolvente o meno. Inoltre, si parla di soluzioni non-custodial, grazie alla realizzazione delle transazioni tramite smart-contract.
Va detto che questo non è l'unico modo per offrire prestiti sfruttando le criptovalute: sono anche presenti piattaforme di Crypto Lending centralizzate che si propongono di massimizzare gli interessi per chi deposita. Tuttavia, in questo scenario si ricreano tutte le problematiche di custodia dei beni e di privacy per gli utenti che scelgono di usufruire di tali servizi. Pertanto si può dire che una piattaforma di Crypto Lending centralizzata non è tanto diversa da una banca. Ciò che può sembrare strano, ad ora, è la questione della collateralizzazione del

prestito, in quanto è necessario assicurare con una quantità di denaro generalmente superiore la cifra che si chiede in prestito. Lo scopo di un prestito di questo tipo è quello di mantenere l'esposizione sul mercato delle criptovalute, in grande crescita, mantenendo comunque la possibilità di spendere il proprio denaro, ottenendo delle stablecoin. Molto spesso però è necessario impegnare una cifra superiore a quella che si intende investire ed è nuovamente dovuto all'estrema variabilità delle criptovalute impegnabili: un'*extracollateralizzazione* del prestito copre la piattaforma da un'eventuale insolvenza del debitore dovuta magari ad un crollo delle criptovalute.

Prestiti con lending pool

Fino ad ora abbiamo parlato di prestiti effettuati accoppiando domanda e offerta, in maniera molto simile a quanto fatto dagli exchange che si servono di un order book. Tuttavia, nel caso dei prestiti, risulta più difficile trovare una corrispondenza fra domanda e offerta. Questo è stato il più grande problema che la piattaforma di prestiti ETHlend [7] si trovò ad affrontare e che ha portato alla creazione di AAVE, una piattaforma che, al contrario, adotta una strategia del tutto analoga a quella usata dagli AMM. Come ogni AMM, essa consiste in un insieme di liquidity pool (che in questo caso viene chiamata lending pool), dove gli utenti possono decidere di depositare le loro disponibilità di liquidi (e quindi ricevere un tasso di interesse in cambio) e chiunque necessiti può prendere in prestito del denaro mettendo come garanzia criptomonete. Gli interessi pagati vengono stabiliti in base alla quantità di denaro presente nella pool stessa.

Presiti non collateralizzati

Oltre ai prestiti di tipo secured, l'obiettivo della DeFi è anche quello di offrire prestiti senza garanzia sul denaro richiesto. Tradizionalmente, le banche offrono questo servizio (anche se, dopo il crollo di Lehman Brothers è molto facile che venga richiesta una copertura per il prestito richiesto) a persone che quasi certamente saranno in grado di ripagare il prestito, con un controllo sulla storia bancaria di chi contrae il debito, sulle capacità economiche e sulle motivazioni. In seguito, si elaborerà un punteggio di affidabilità che determinerà se concedere il prestito o meno. Le piattaforme di DeFi sono ancora agli albori e non esistono al momento strumenti per valutare l'affidabilità creditizia di una persona. Di fatto, ciò fa sì che un sistema di prestiti unsecured, come quello offerto dalle

banche, non sia ancora presente. Nonostante ciò, si è comunque provato a muovere i primi passi verso questo tipo di prestiti con l'introduzione dei flash loans. I flash loans possono essere definiti come prestiti che vengono erogati e ripagati nella stessa transazione. Si può pensare al programma della transazione come suddiviso in tre parti: ricezione del credito, operazioni con il credito, rimborso del credito. Il prestito va a buon fine se il rimborso è effettuato entro il lasso di tempo prescritto dalla transazione. In caso contrario, il network rifiuta la transazione e i fondi tornano al prestatore. Tale modalità permette di annullare il rischio di insolvenza da parte di chi contrae il credito poiché in caso di insolvenza lo smart contract effettuerà in modo automatico il rimborso. Sebbene sia una modalità ancora molto limitata, essa permette di attuare la pratica dell'arbitraggio, anche con guadagni e speculazioni potenzialmente molto grandi (https://academy.binance.com/it/articles/what-areflash-loans-in-defi). Da un lato, dunque, i flash loans sono uno strumento che aumenta l'inclusione finanziaria del sistema, poiché permettono di ottenere dei prestiti anche ad utenti che non dispongono di grandi quantità di criptomonete. Dall'altro, essi permettono vere e proprie manipolazioni del mercato da parte di utenti con grandi disponibilità economiche.

Uniswap

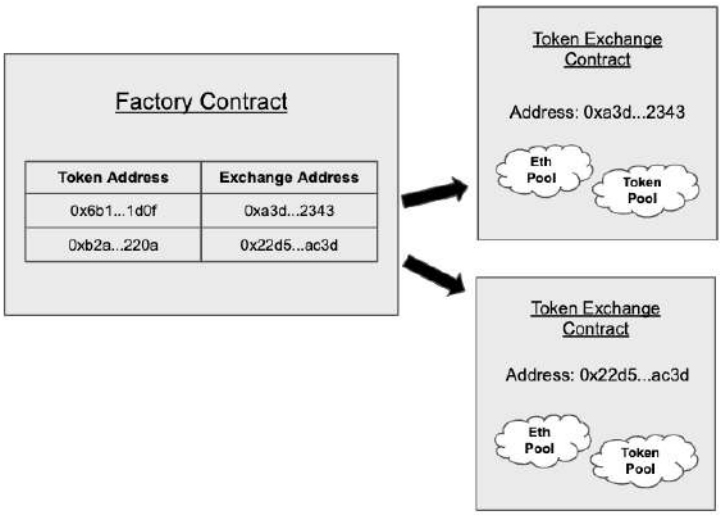
In questa sezione andiamo ad analizzare nel dettagio uno specifico DEX. Il nostro caso studio si chiama **Uniswap**, protocollo di scambio decentralizzato basato sulla blockchain di Ethereum che sfrutta i tokens ERC-20 e che regola gli scambi mediante smart contract. [9]

Come funziona uno scambio su Uniswap

Uniswap ricade nella categoria degli **AMM**. In tale protocollo vi sono dunque dei liquidity provider che depositano, in un'ottica di guadagno, i propri tokens all'interno di una liquity pool da cui gli utenti che desiderano scambiare i propri token (Ether con ERC-20 oppure scambi tra token ERC-20) possono attingere pagando delle fee (che vengono decise in base alla disponibilità dei tokens richiesti nel pool), le quali verranno poi suddivise tra i liquidity provider. Il valore dei token in una liquidity pool cambia continuamente, in base alla quantità presente nella pool. Come già specificato in precedenza, le liquidity pool possono essere soggette a pratiche di arbitraggio da parte di alcuni utenti. Tuttavia, come peraltro specificato nella documentazione di Uniswap, l'arbitraggio è, entro certi limiti, ben visto, in quanto permette un riallineamento automatico del prezzo dei token all'interno della pool con quello del mercato esterno, garantendo il corretto funzionamento della pool. Infine, nella nuova versione del protocollo, il valore dei beni depositati dai liquidity provider non varia più tra 0 e infinito, ma deve avere un intervallo fissato, quello che secondo il provider genererà il maggior numero di fee e quindi maggior profitto.

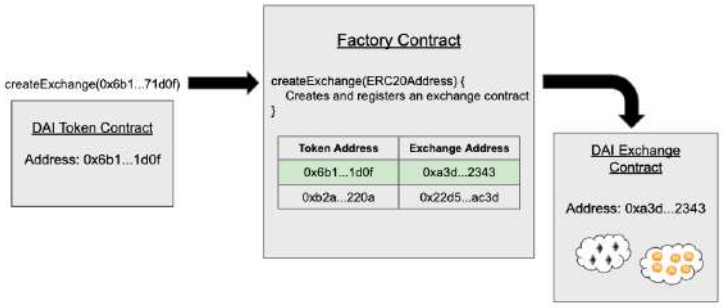
I contratti su Uniswap

In precedenza, si è detto che i liquidity provider mettono a disposizione dei fondi che vengono congelati grazie agli smart contracts. Più precisamente, affinché sia possibile lo scambio, sono necessari dei set di contratti sia per la creazione delle liquidity pool che per lo scambio dei token di tipo ERC-20. In un certo senso, si può affermare che Uniswap si occupa di standardizzare questo processo per agevolare le transazioni. I contratti di Uniswap possono essere raggruppati in due diverse tipologie: exchange contract e factory contract. Il primo tipo di contratto si occupa di creare un pool di token ed ETH con cui gli utenti possono interagire per scambiare il denaro (che ricordiamo può avvenire tra token ERC20 diversi o tra token ERC-20 e ETH). Il secondo tipo si occupa di creare gli exchange contracts e di assegnare loro l'indirizzo dei token.

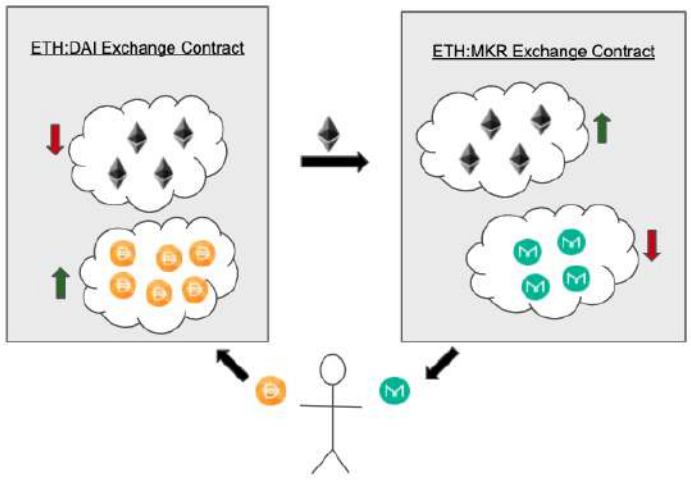


E' interessante notare come non ci siano dei vincoli sulla tipologia di token utilizzabili come mezzo di scambio. Questo, in generale, non è garantito dai CEX, i quali, tendenzialmente, permettono solamente scambi di token per cui ci sia un alto volume di domanda e offerta. In quanto sistema decentralizzato, Uniswap permette invece di trattare un qualunque tipo di token. L'utente deve solamente richiamare una funzione del Factory Contract per registrare un nuovo token ed aggiungere l'indirizzo di token associato ad un indirizzo di scambio,

producendo poi un nuovo exchange contract. Tale procedimento è raffigurato nell'immagine sotto riportata.



Aggiungere un token su Uniswap è, peraltro, gratuito. Infine, quando si tratta di uno scambio tra Token ERC-20 differenti, viene chiamata la funzione tokenToTokenSwap, che scambia il denaro per l'utente che ne fa richiesta e si occupa di scambiare ETH tra i pool. Possiamo quindi concludere che tutti i pool su Uniswap sono delle coppie token-ETH, che possono interagire tra di loro se l'utente ne ha necessità.



Il fallimento di una transazione

Il completamento di una transazione non è sempre garantito. Può capitare, infatti, che il gas necessario per registrare la transazione sulla blockchain non sia sufficiente e quindi la transazione può restare in attesa per un tempo indefinito, permettendo anche che altre transazioni le passino avanti. Questo però può essere un problema, in quanto ad ogni scambio corrisponde una variazione del prezzo del bene nella pool. Se al momento dell'effettiva esecuzione della transazione il prezzo del token supera la percentuale di tolleranza definita dall'utente al momento della sottomissione allora la transazione non viene eseguita: questa dinamica è detta *slippage*.

Perdita impermanente

Come detto prima, il liquidity provider nella v3 di Uniswap può decidere l'intervallo di prezzo entro il quale può variare la liquidità messa a disposizione nel *pool*. Supponiamo di trascurare per ora le commissioni e concentriamoci solo sul prezzo che hanno i beni nella pool che ricordiamo mantenere costante il prodotto:

$$\#Eth * \#token_A = k$$

e chiameremo **k** la **liquidità** del pool. Prendiamo ora un caso in cui 1ETH nel mercato valga 100 DAI e supponiamo che ci sia un pool con 10 ETH e 1000 DAI, con una liquidità pari a 10000. Supponiamo ora che Alice abbia depositato in questo pool 1 ETH e 100 DAI, quindi detiene il 10% del pool e la sua posizione, al momento del deposito, vale 200 DAI. Immaginiamo ora una situazione in cui il valore di ETH quadruplica e arriva a valere 400 DAI, creando quindi uno squilibrio del valore della pool rispetto al mercato esterno. L'arbitraggio, in questo caso, è molto semplice, in quanto è sufficiente pagare 1000 DAI per prelevare 5 ETH, che potrò vendere nel mercato per ottenere un guadagno netto di 1000 DAI. All'interno della pool invece, il valore di ETH ora riflette quello del mercato reale: ci saranno 5 ETH e 2000 DAI. Immaginiamo ora che Alice voglia ritirare i fondi depositati che corrispondono 10% dei beni depositati nella pool, ottenendo 200 DAI e 0.5ETH, per un valore di 400 DAI, più alto rispetto al deposito. Tuttavia, se Alice non avesse depositato e avesse tenuto (in gergo *HODLato*) i suoi asset, il valore del suo portafoglio sarebbe ora

pari a 500 DAI. Questa discrepanza viene chiamata *impermanent risk* o **perdita impermanente** ed è una perdita che viene realizzata solo nel momento in cui il provider decide di ritirare il proprio denaro nel pool. [10]
La nuova versione di Uniswap permette di mitigare questo problema. Abbiamo visto che gli arbitraggiatori mantengono il prezzo nel pool simile a quello del mercato esterno, mentre chi ci rimette in questo caso è il *liquidity provider*. Avendo introdotto la possibilità di scegliere l'intervallo di prezzo entro il quale i propri liquidi forniti possono essere scambiati, si possono limitare di molto le perdite dovute all'impermanent risk, in quanto i beni che escono dall'intervallo fissato dal provider non possono essere più scambiati ma vengono congelati nella pool. Se il provider non ritira il proprio deposito esso verrà sbloccato nel momento in cui il valore rientra nell'intervallo fissato. Alla luce di questo è importante osservare che il liquidity provider ha tutto l'interesse di depositare, all'interno della pool, una quantità di token tale che rispecchi il loro valore rispettivo nel mercato, per evitare di creare immediate possibilità di arbitraggio.
Una seconda questione ignorata nell'esempio di prima ma comunque rilevante è la questione delle *fee*, che vengono suddivise equamente tra tutti i liquidity providers della pool. Con la nuova versione di *Uniswap* sono presenti 3 opzioni di fee: 0.05%, 0.3% e 1%. L'esistenza delle fee serve a incoraggiare i liquidity providers in quanto permettono di fronteggiare l'impermanent risk, fornendo una sorta di cuscinetto in caso di piccole variazioni del prezzo dei beni. E' chiaro che, maggiore sarà la fee (che è decisa dal liquidity provider), maggiore sarà la volatilità dei beni nella pool.

La perdita impermanente nella realtà

E' interessante anche notare quanto nella realtà questa perdita impermanente conti. Abbiamo detto che un generico liquidity pool di Uniswap rispetta questa formula

$$\#ETH * \#token_A = liquidity$$

e possiamo definire il prezzo di eth come

$$prezzo_{ETH} = \frac{\#token_A}{\#ETH}$$

da cui possiamo ricavare la quantità di ethereum nella pool rispetto al suo prezzo

$$\#ETH = \sqrt{\frac{liquidity}{prezzo_{ETH}}}$$

e possiamo fare un calcolo simile per il prezzo del token, da qui otteniamo

$$\#token_A = \sqrt{liquidity * prezzo_{ETH}}.$$

Possiamo anche derivare dalla formula sopra una quantità chiamata *divergence loss*, che è pari a

$$divergence = 2\frac{\sqrt{R}}{1+R} - 1, \text{ con } R = \frac{prezzoiniziale_{ETH}}{prezzoattuale_{ETH}}$$

con cui possiamo vedere l'impermanent risk in base alla variazione di prezzo rispetto al momento del deposito.

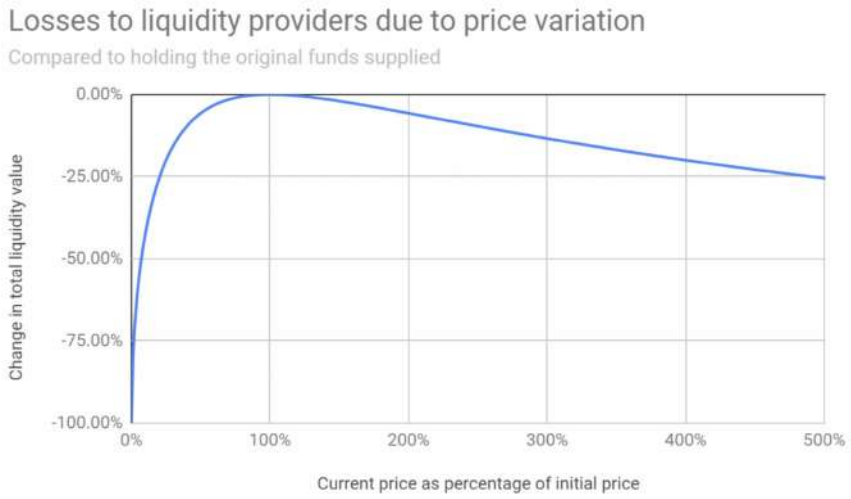


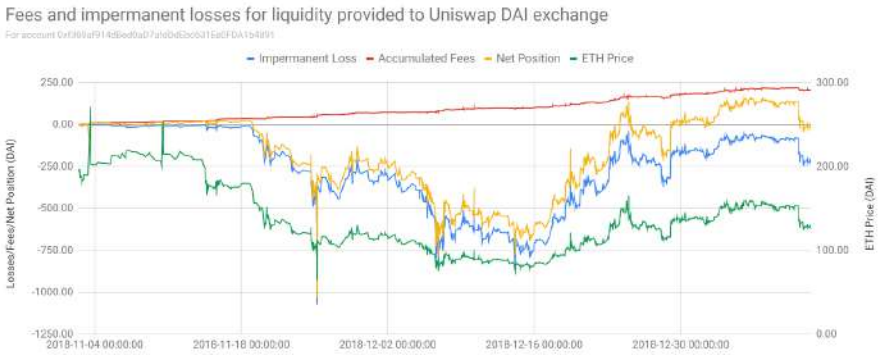
Grafico della divergence loss

Rispetto alla strategia HODL possiamo dunque vedere le perdite percentuali:

- una variazione del prezzo di 1.25x comporta una perdita relativa ad HODL pari allo 0.6%;

- una variazione del prezzo di 1.5x comporta una perdita relativa ad HODL pari allo 2%;
- una variazione del prezzo di 1.75x comporta una perdita relativa ad HODL pari allo 3.8%;
- una variazione del prezzo di 2x comporta una perdita relativa ad HODL pari allo 5.7%;
- una variazione del prezzo di 3x comporta una perdita relativa ad HODL pari allo 13.4%;
- una variazione del prezzo di 4x comporta una perdita relativa ad HODL pari allo 20%;
- una variazione del prezzo di 5x comporta una perdita relativa ad HODL pari allo 25.5%;

ed è interessante osservare come non sia importante la variazione del prezzo (in positivo o in negativo). Abbiamo detto però che le fee mitigano questo effetto e un esempio è dato dall'account *0xf369af914dBed0aD7afdDdEbc631Ee0FDA1b4891* [11] che ha iniziato con un deposito pari a 30 ETH e 5900 DAI a Novembre 2018.



La curva che ci interessa è quella gialla, relativa alla posizione netta, e si può vedere che nonostante il prezzo di ETH sia variato (e quindi il proprietario avrebbe totalizzato una perdita impermanente se avesse ritirato la propria liquidità), le fee hanno permesso di mitigare questo fatto, portando a una posizione netta positiva alla fine di Novembre 2018.

AAVE

Analizziamo ora un protocollo decentralizzato per la concessione di prestiti chiamato **AAVE**. Come detto nell'introduzione, stiamo analizzando protocolli di tipo non-custodial, dove gli utenti rimangono in possesso del denaro che depositano. Inoltre, chiunque può partecipare come fornitore di liquidità o come prestatore. Il protocollo AAVE è l'evoluzione dell'ormai defunto ETHLend. Tale sistema ad oggi muove intorno ai 17 miliardi di dollari ed è uno dei protocolli di prestiti decentralizzati più famosi. A differenza di ETHLend, che accoppiava chi offriva e chi richiedeva liquidità, AAVE sfrutta gli smart contracts per creare dei *lending pool* da cui gli utenti possono attingere il denaro di cui necessitano, ad un tasso di interesse che dipende dalla liquidità disponibile nella pool.

Chi deposita, invece, riceve dei *aTokens* che tengono traccia dei soldi depo-

sitati e dei relativi tassi di interesse che gli spettano. Possiamo quindi definire gli *aTokens* come certificati che vengono coniati al momento del deposito e che sono ancorati 1:1 al valore dell'asset depositato . Questi, in seguito, potranno essere scambiati anche su altri protocolli o DEX.

Come funziona un prestito su AAVE

I prestiti concessi su AAVE [12] (che generalmente indicano la valuta presa in prestito in dollari) vengono erogati con una stablecoin a scelta dell'utente (USDC, DAI, USDT etc.). I tassi di interesse variano a seconda del mercato. All'utente vengono offerte due opzioni: tasso fisso e tasso puramente variabile. Un tasso di interesse puramente variabile si basa unicamente sulla domanda e sull'offerta del *pool*, mentre un tasso di interesse stabile agisce come un tasso di interesse fisso in un determinato periodo di tempo (nel nostro caso è un giorno). Va tenuto conto che, in generale, uno stable rate ha un costo più alto rispetto ad un variable rate. Tuttavia, esso garantisce protezione in caso di repentine variazioni del valore del bene preso in prestito, che in questo caso dipende dall'utilization rate nel pool.

Ad ogni prestito viene associato un indicatore chiamato health factor: esso indica quanto gli asset depositati come garanzia siano sicuri in relazione all'ammontare dei prestiti ottenuti e al loro valore attuale.

$$H_f = \frac{TotalCollateralETH * L_Q^a}{TotalBorrowETH + TotalFeesETH}$$

dove L_Q^a è la soglia di liquidazione media.
Affinché non venga effettuata la liquidazione del prestito (la somma prestata è automaticamente restituita al proprietario rifacendosi sui collaterali) l'health factor non deve scendere al di sotto di 1. Per tale ragione, quando si prende in prestito una determinata somma su AAVE, si tende ad *extracollateralizzare* il prestito, cioè mettere a garanzia una somma superiore rispetto a quella presa in prestito, così da evitare la liquidazione automatica. È sempre possibile depositare nuovi asset in modo da aumentare l'health factor. È interessante notare come all'erogazione di un prestito non corrisponda una data fissata per la restituzione. Questa, a meno del caso in cui il prestito non venga liquidato automaticamente, può essere decisa da chi ha contratto il debito. Si osservi però che, con il passare del tempo, cresce l'interesse da pagare e dunque l'health fac-

tor scende di conseguenza. L'erogazione del prestito è dunque condizionata alla garanzia dello stesso. Infine, si noti che il rischio di insolvenza è praticamente annullato, visto il processo di liquidazione automatica.

Rischio della valuta o Currency risk

I rischi di insolvenza, innanzitutto grazie all'*extracollateralizzazione* dei prestiti, sono sicuramente molto bassi ma non nulli, a causa dell'estrema variabilità delle criptomonete che possono essere messe a garanzia. Per far fronte a variazioni in negativo troppo grandi nel valore dei collaterali depositati dai propri utenti, i creatori del protocollo hanno deciso di accettare come garanzia del prestito solo alcune criptomonete attentamente selezionate. Di fatto, la diversificazione delle monete presenti sulla piattaforma fa fronte al rischio di fallimento, mitigandolo, e permette anche di stabilizzare il valore delle pool in momenti di estrema volatilità. La scelta delle monete da accettare, tuttavia, deve essere fatta in modo che il rischio di insolvenza sia minimizzato e si mantenga in ogni caso decentralizzato, vengono escluse quindi monete centralizzate che potrebbero introdurre un single point of failure. Affinché una moneta possa essere accettata è opportuno pertanto che sia un progetto con grandi possibilità di espansione (Ethereum è un esempio) supportato da una vasta community così che i rischi di default siano minimizzati. Sul sito di AAVE sono elencate tutte le criptomonete accettate come garanzia.

Tassi di interesse su AAVE

Ogni prestito per essere erogato necessita di definire dei tassi di interesse che ripaghino il creditore. AAVE implementa la possibilità di scegliere tra due differenti tassi di interesse[13]: quello variabile e quello detto "stable". Il secondo, in generale, presenta dei costi maggiori, ma offre una maggiore copertura dei rischi. Inizialmente viene calcolato il fattore U, detto utilization rate, per ogni moneta. Esso rappresenta la quantità di denaro prestata rispetto alla quantità di denaro presente. Per ogni asset viene definita la quantità ottimale $U_{optimal}$ di importante utilizzo per il calcolo dei tassi di interesse. Lo storico dell'utilizzo di AAVE mostra come ogni asset ha un profilo di rischio di liquidità che evolve con le variazioni del mercato e dunque non rimane costante. In particolare, il

tasso di interesse variabile si ottiene dalla seguente relazione:

$$R_v = \begin{cases} R_{v_0} + \frac{U}{U_{\text{optimal}}} R_{\text{slope 1}}, \text{ if } U \leq U_{\text{optimal}} \\ R_{v_0} + R_{\text{slope 1}} + \frac{U - U_{\text{optimal}}}{1 - U_{\text{optimal}}} R_{\text{slope 2}}, \text{ if } U > U_{\text{optimal}} \end{cases}$$

in cui

- R_{v_0} è il tasso di interesse variabile di base
- R_{slope1} è la pendenza del rate di interesse sotto $U_{optimal}$
- R_{slope2} è la pendenza del rate di interesse sopra $U_{optimal}$

che quindi mostra come una volta superato il valore $U_{optimal}$ dell'utilization rate il tasso di interesse cresca molto più rapidamente. Per quanto riguarda il tasso di interesse "stable", esso non è fissato durante tutta la durata del prestito, ma, come ci si può aspettare, risente delle variazioni di mercato. Esso viene quindi modificato con una certa cadenza (un giorno). Dunque, al contrario del tasso di interesse variabile, i processi di calcolo e di riaggiustamento del tasso stesso non sono svolti in modo continuo. In questo modo, chi ottiene il prestito non rischia di vederlo ritirato per diminuzioni improvvise dell'Health factor. Il calcolo effettivo di tale interesse è molto simile a quello precedente ed è il seguente:

$$R_s^t = \begin{cases} M_r + \frac{U}{U_{\text{optimal}}} R_{\text{slope1}}, \text{ if } U \leq U_{\text{optimal}} \\ M_r + R_{\text{slope 1}} + \frac{U - U_{\text{optimal}}}{1 - U_{\text{optimal}}} R_{\text{slope 2}}, \text{ if } U > U_{\text{optimal}} \end{cases}$$

in cui i parametri sono:

- M_r lending rate medio nel mercato calcolato attraverso un oracolo;
- R_{slope1} è la pendenza del rate di interesse sotto $U_{optimal}$;
- R_{slope2} è la pendenza del rate di interesse sopra $U_{optimal}$.

Un esempio pratico di calcolo dei rate variabile e stabile è dato dal grafico in figura 1. Concordemente a quanto detto in precedenza, si osserva che il tasso ha due pendenze diverse prima e dopo il valore ottimale dell'utilization rate e che il tasso di interesse stabile è maggiore di quello variabile. Ora, come si inseriscono i tassi di interesse nel protocollo di AAVE? Nel momento in cui le monete depositate vengono prese in prestito da altri, il network di AAVE crea

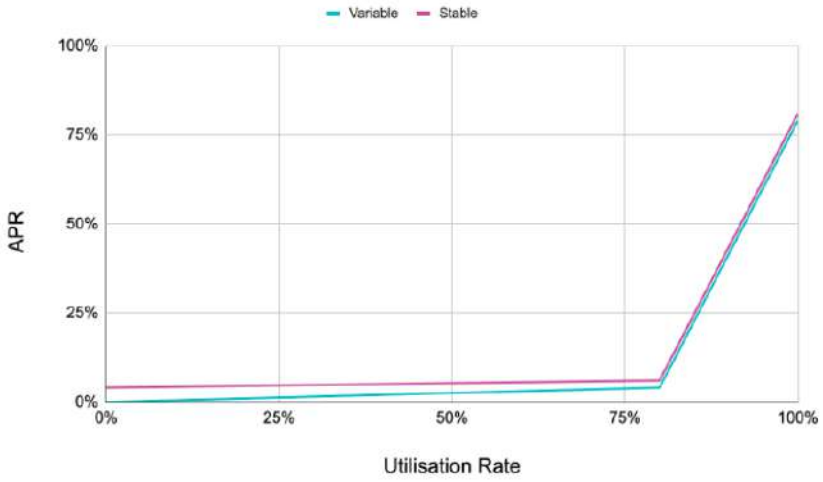


Figura 1: Tassi di interesse

i cosiddetti aToken. Se ad esempio di prestano 500 DAI, una volta immessi nel pool si riceveranno 500 aToken Dai (o aDai), che danno diritto alla maturazione dell'interesse, calcolato nel modo prescritto sopra. Il prezzo di un *aToken* può essere approssimato come il valore dei flussi di denaro (moltiplicati per un fattore di sconto) osservati tra il momento della valutazione e la fine del periodo di riferimento. Il fattore di sconto tiene conto dell'*annual percentual yield* che è maggiore del *risk free rate*. Gli aToken quindi hanno un valore maggiore rispetto a quello del token sottostante.

Cerchiamo di fare chiarezza. Gli aToken rappresentano un asset depositato su AAVE: essi generano interessi attivi e possono essere riscattati in qualsiasi momento in rapporto 1:1 con la moneta sottostante, se disponibile. Questo scenario apre le porte ad un serio rischio: un massiccio riscatto di aToken potrebbe portare a una drastica diminuzione di liquidità mettendo in ginocchio la piattaforma. Per scongiurare questo rischio si è deciso di rendere possibile lo scambio di aToken su exchange e piattaforme di token swap esterne. In questo modo chi vende ottiene un ricavo immediato rappresentato dal risk free rate cioè il valore attualizzato degli interessi che essi generano, i quali saranno incassati dal nuovo possessore dei suddetti token. Ovviamente gli aToken scambiati al di fuori di AAVE possono comunque essere sempre riconvertiti nella moneta originale. In tutto ciò AAVE non perde liquidità e può continuare la sua attività dato che i

fondi depositati non lasciano mai la piattaforma. Gli aToken hanno quindi un valore superiore rispetto a quello delle monete sottostanti per cui gli utenti sono incentivati a tenerli da parte e a non riscattarli. Questo valore si ripercuote sul prezzo di mercato, il quale deve tener conto anche dell'APY, interesse maturato sulla somma originariamente depositata e sugli interessi ottenuti (i quali vengono riconosciuti in tempo reale attraverso un flusso costante di moneta nei confronti del loro possessore). Per illustrare questo fenomeno prendiamo come esempio gli aDAI, aToken associato alla stablecoin DAI.

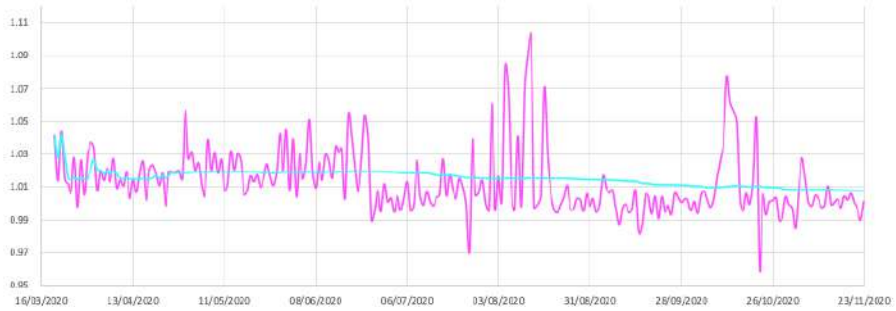


Figura 2: Prezzo di aDai in Dai

1 aDAI è riscattabile per 1 DAI in qualsiasi momento ma il loro prezzo di mercato, come si può vedere in figura 3, è comunque volatile. Il grafico ci mostra come la mediana sia pari a 1 DAI = 1.01 aDAI confermando quindi quanto detto in precedenza, cioè che gli aToken hanno un valore maggiore dei token sottostanti.

La strategia di gestione dei tassi d'interesse di AAVE permette dunque di avere una costante presenza di asset da poter concedere in prestito senza incorrere in rischi di liquidità incoraggiando gli utenti a conferire fondi nella pool. In particolare, si ha che quando la disponibilità di capitale è alta i tassi d'interesse sono ridotti in modo da incoraggiare i prestiti; al contrario se la disponibilità di fondi è bassa i tassi d'interesse subiscono un incremento in modo da incentivare nuovi depositi e spingere i debitori a ripagare i loro debiti.

Bibliografia

[1] DeFi Pulse
https://defipulse.com

[2] Il tweet di un utente comune su twitter.
https://twitter.com/staking_gwei/status/1190192003474169856

[3] Che cos'è un exchange decentralizzato?, Binance Academy
https://academy.binance.com/it/articles/what-is-a-decentralized-exchange-dex

[4] Che cos'è un AMM?, Binance Academy
https://academy.binance.com/it/articles/what-is-an-automated-market-maker-amm

[5] Pools, Uniswap Documentation
https://uniswap.org/docs/v2/core-concepts/pools/

[6] Fintech lending Industry to hit USD 390.5 billion by 2023, The Paypers
https://thepaypers.com/payments-general/fintech-lending-industry-to-hit-usd-3905-billion-by-2023--1240552

[7] ETHLend website
https://ethlend.io

[8] Cosa sono i *flash loans* nella finanza decentralizzata?, Binance Academy
https://academy.binance.com/it/articles/what-areflash-loans-in-defi

[9] La documentazione di Uniswap V3
https://docs.uniswap.org

[10] Uniswap: A Good Deal for Liquidity Providers?, Medium
https://pintail.medium.com/uniswap-a-good-deal-for-liquidity-providers-104c0b6816f2

[11] Etherscsan for addess 0xf369af914dBed0aD7afdDdEbc631Ee0FDA1b4891
`https://etherscan.io/address/0xf369af914dbed0ad7afdddebc631ee0fda1b4891`

[12] Documentazione per gli sviluppatori di AAVE, AAVE
`https://docs.aave.com/developers/`

[13] Whitepaper relativo ad AAVE
`https://github.com/aave/protocol-v2/blob/master/aave-v2-whitepaper.pdf`

VeChain

UNA BLOCKCHAIN ENTERPRISE FOCUSED

Simonetta Bodojra, Luca Cataldo, Giulio Cerruto, Alessio Claudio

Indice

Capitolo 1

Introduzione

`VeChain` è un progetto su blockchain che punta ad offrire strumenti alle grandi imprese, principalmente per il tracciamento della filiera e dei propri prodotti. Si può considerare un progetto principalmente *enterprise focused*, che punta ad essere un ecosistema dove le aziende e gli sviluppatori possono trovare tutto il necessario per realizzare applicazioni, tracciamenti di filiera e sistemi di controllo.

`VeChain` integra principalmente tre diverse funzionalità sul suo network:

- Una blockchain multilayer;
- Un toolkit per lo sviluppo di app decentralizzate;
- Un hardware compatibile sviluppato dalla società stessa.

L'ultima funzionalità è una particolarità di `VeChain` che permette di implementare progetti che, ad esempio, possono tracciare l'originalità di un prodotto o che possono seguire la vita meccanica ed elettrica di un'auto, cosa che è stata già implementata da brand come BMW.

La scelta di `VeChain` di svilupparsi relativamente alle necessità delle aziende è strettamente legata al loro obiettivo finale: dell'adozione di massa della tecnologia della blockchain. In generale, quando una nuova tecnologia arriva sul mercato la sua diffusione dipende dalla volontà di adozione dei consumatori, che possono essere divisi in varie categorie (vd. Figura 1):

- Innovators: sono persone che lavorano in ambito tecnologico che sono in grado di dare dei feedback per migliorare la nuova tecnologia;
- Early adopters: sono utilizzatori che entrano in contatto con la tecnologia per motivi lavorativi o di passione che spingono la diffusione iniziale;
- Early majority: sono la maggioranza dei consumatori, persone non sono per forza esperte di tecnologia, disposte a provare qualcosa di nuovo.

Il salto dagli early adopters all'early majority si dice "crossing the chasm", ed è effettivamente il passaggio più difficile per ogni tecnologia, che decreta se effettivamente questa sarà in grado di diffondersi o meno. Per quanto riguarda la tecnologia della blockchain, gli innovators possono essere identificati in coloro che hanno contribuito a sviluppare le dApps e a minare le iniziali PoW, mentre gli early adopters sono coloro che hanno iniziato a creare vari progetti con i primi casi applicativi. In generale però, il consumatore medio ancora non sa cosa sia la blockchain e perciò, `VeChain` sostiene che per superare il "Chasm" e riuscire a raggiungere una adozione di massa, il percorso da intraprendere non sia quello della DeFi o di Tesla che investe milioni in Bitcoin, ma sia necessario convincere l'early majority ad adottare questa tecnologia. Un modo per farlo è, per esempio, trovare una nicchia di mercato che permetta alla blockchain di crescere e fornire servizi sempre più personalizzati, infatti come nessuna compagnia di automobili ha il 100% di market share, `VeChain` sostiene che nessuna blockchain potrebbe essere efficace in ogni ambito possibile, perciò ha deciso di essere *enterprise focused*.

L'obiettivo dell'adozione di massa ha portato `VeChain` ad avere una struttura particolare, non solo della sua blockchain ma soprattutto del il suo modello economico e della sua governance, come si vedrà nei prossimi capitoli.

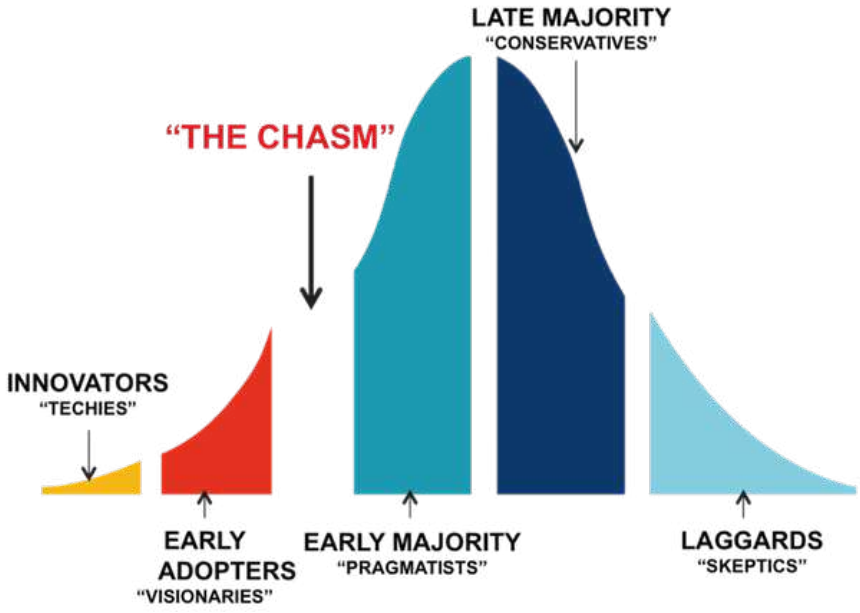


Figura 1.1: **Curva di Moore.** Diagramma proposto da Geoffrey A. Moore per descrivere il fenomeno della diffusione e dell'adozione di una nuova tecnologia, dividendo i consumatori in base alla loro volontà di adozione.

Capitolo 2

La blockchain VeChainThor

Si parte quindi da una blockchain pubblica, chiamata VeChainThor, che ha come obiettivo quello dell'adozione di massa, con un focus specifico in termini di piccole, medie e grandi imprese. Il servizio era nato inizialmente all'interno del progetto Ethereum, per poi spostarsi su una rete propria nel 2018, per due principali motivi:

1. Ethereum non ha una vera a propria strategia di consenso per permettere transazioni trasparenti e efficienti;

2. Ethereum non ha un modello economico adatto per permettere alle aziende di avere delle dApps con un costo prevedibile, infatti considerando la volatilità di ETH, è quasi impossibile per le aziende predire i costi.

2.1 Proof of Authority

VeChainThor utilizza un algoritmo di consenso chiamato *Proof-of-Authority* (PoA). Questo modello si basa su un numero limitato di validatori che devono rendere pubblica la loro identità ed avere l'autorizzazione da parte della Fondazione per partecipare al consenso della blockchain. Questi nodi pre-autorizzati vengono chiamati *Authority Masternodes* (AM) e il loro numero è fissato a 101.

Gli AMs guadagnano il diritto di diventare validatori e sono incentivati a mantenere questa posizione proprio perchè la loro identità è svelata. Infatti, attribuendo una reputazione all'identità, i validatori sono incentivati a sostenere il processo di transazione in quanto non desiderano che la loro identità sia collegata a una reputazione negativa. Di conseguenza, a differenza della PoW, non è necessario che i nodi impieghino una grande quantità di risorse per competere tra loro o che, a differenza della PoS, i nodi più ricchi non hanno più vantaggi di altri nodi nel sistema.

Il numero limitato di validatori, permette di avere alta scalabilità, cioè di aggiornare la blockchain più frequentemente riducendo il tempo tra ogni blocco ed elaborare più transazioni per commissioni di elaborazione vicine allo zero.

Questo sistema gioca a sfavore della decentralizzazione del sistema, ma rimane un modello coerente con l'idea della VeCahin Foundation: *"Neither a total centralization nor a total decentralization would be the correct answer, but a compromise from and balance of both would"*

La PoA utilizza una regola per determinare la catena canonica, chiamata, "trunk" in caso di fork: si computa l'accumulated witness number (AWN) π_i, del blocco i con la seguente formula:

$$\pi_i = \pi_{i-1} + A_i$$

con $\pi_{genesis} = 0$. Il secondo termine calcola il numero di AMs "attivi", cioè il numero di nodi di consenso che sono testimoni della generazione del blocco i. Gli altri nodi dovranno concordare il valore di π_i per accettare il blocco e memorizzarlo come `TotalScore`. Un nodo verrà contrassegnato come "inattivo" da altri nodi dopo che non ha generato il nuovo blocco nel round mentre era il leader. Un nodo inattivo verrà considerato "attivo" una volta che produrrà nuovamente un nuovo blocco. In caso di una fork quindi, si sceglierà il blocco con il `TotalScore` maggiore, accumulato sin dal blocco di genesi, cioè prediligerà il ramo su cui hanno lavorato più AMs.

Un possibile difetto della PoA è che, essendo i validatori piuttosto pochi e conosciuti, un attaccante potrebbe cercare di corromperli per compromettere il sistema dall'interno, bisognerebbe quindi dissuadere il nodo stesso dalla manipolazione del sistema quando gli viene dato il diritto di aggiungere un nuovo blocco. Inoltre, anche PoA fornisce solo una garanzia probabilistica della sicurezza delle transazioni, come tutti gli algoritmi di consenso di Nakamoto, che potrebbe non essere sufficiente per mantenere la coerenza del sistema in caso di una fork temporanea. Per queste ragioni, `VeChain` ha proposto una nuova versione della PoA.

2.1.1 PoA 2.0 (SURFACE)

Coerentemente con l'idea dell'adozione di massa, particolare attenzione è stata rivolta a migliorare ulteriormente la scalabilità dell'algoritmo di consenso, quindi aumentare il numero di transazioni validate per secondo. Ci sono molti modi per migliorare la scalabilità di una blockchain, come aumentare la capacità di trasmissione (bandwidth) o aumentare la grandezza di un blocco o la frequenza con cui questo viene mandato. Fissare gli AMs a 101 vuol dire fissare la capacità del network, quindi, su `VeChainThor`, la bandwidth risulta essere un po' il collo di bottiglia e non può essere migliorata in nessun modo. Per questo motivo si è scelto di ottimizzare il suo utilizzo con diverse tecniche spiegate in seguito.

Come si può vedere nell'immagine 2.1.1, quando un blocco viene creato, nella PoA tradizionale il tempo viene utilizzato per comunicare le generalità del blocco (parte nera), per trasmettere (parte blu) e validare il blocco (parte a righe) e infine per raggiungere il consenso (parte grigia). Per ottimizzare al meglio un round, e di conseguenza l'utilizzo della bandwidth, l'ideale sarebbe aumentare il tempo di trasmissione il più possibile.

Con questo obiettivo, PoA 2.0 propone le seguenti tecniche:

- **Delayed validation**: invece che validare il blocco nel round in cui viene generato, la validazione è ritardata al round successivo, sostanzialmente al round r viene validato il blocco generato al round $r-1$. In questo modo la validazione e la trasmissione dei blocchi può essere fatta in parallelo, risparmiando il tempo per la validazione (parte a righe). Nella PoA 2.0 il consenso avviene tramite un comitato e un leader, come si vedrà nella pagina successiva, e se il comitato validasse la transazione generata nello stesso round r, allora il leader dovrebbe aspettare la conferma da parte di tutti i membri, perdendo sostanzialmente tempo. In questo modo invece, il comitato del round r si occuperà di validare le transazioni già ricevute nel round $r-1$, mentre i membri del comitato stanno ricevendo il blocco per il round corrente r, che verrà a sua volta validato nel round $r+1$;
- **Epoch based random beacon scheduling**: invece di decidere sul momento quale nodo genererà il blocco successivo, un programma viene creato in modo casuale in anticipo per decidere quali, e in che ordine, AMs produrranno i blocchi successivi, per tutta la durata dell'epoca(per esempio un giorno prima). Questo elimina il tempo perso per raggiungere il consenso (parte grigia) ogni volta che un blocco viene creato.

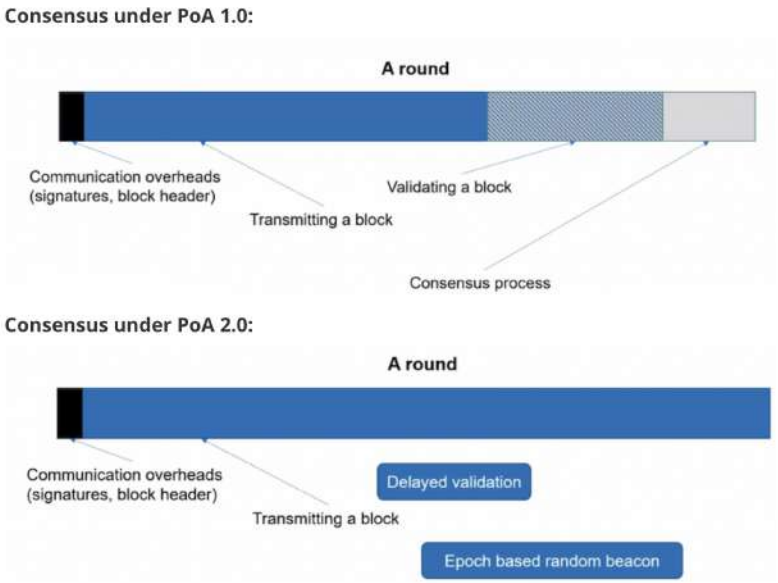


Figura 2.1: **Confronto tra PoA 1.0 e PoA 2.0.**Una delle principali differenze tra la PoA 1.0 e PoA 2.0 è come viene utilizzato il tempo durante la validazione di un blocco. La PoA risparmia infatti sul tempo di validazione e consenso utilizzando due nuove tecniche: la delayed validation e l'epoch based random beacon scheduling.

Quando un nuovo blocco viene creato, ci vuole del tempo affinché i dati del blocco raggiungano tutti gli Autorithy Masternodes e, siccome questi si trovano in luoghi diversi, riceveranno le informazione in momenti diversi. Togliendo dal round il tempo per la validazione e il consenso, i nuovi dati non hanno abbastanza tempo per propagarsi totalmente, portando così a problemi di asincronia tra nodi, aumentando la possibilità di una fork, che sprecherebbe bandwidth e darebbe la possibilità ad un nodo di attaccare la blockchain, riducendone la sicurezza. Una soluzione potrebbe essere impedire a monte la presenza di fork rendendo il consenso di ogni blocco definitivo, utilizzando cioè algoritmi BFT (Bizantine Fault Tolerant), che risolvono il problema di raggiungere il consenso totale in reti dove è possibile avere dei nodi malevoli. Questi tipi di algoritmi però introducono nuovi problemi, infatti consumano tanta bandwith e durano molto, in quanto hanno bisogno di più passi per raggiungere il consenso assoluto. L'approccio della PoA 2.0 si può considerare una via di mezzo tra un algoritmo di consenso di Nakamoto, detto anche "leader-based", dove un nodo della rete ha la facoltà di attaccare il blocco alla catena principale, e tra un algoritmo BFT, che prevede un consenso assoluto per la validazione del blocco, introducendo due meccanismi principali:

1. **Meccanismo di approvazione del comitato**: ad ogni round di consenso, alcuni nodi saranno scelti casualmente, tramite una *verifiable random function* (VRF), come membri del comitato che deve convalidare, insieme a leader, la proposta del blocco e approvare formalmente la proposta firmandola. La grandezza del comitato si adatta alle condizioni del network in modo tale da minimizzare la probabilità di fork. Per creare più blocchi in conflitto, il leader dovrebbe quindi lavorare con un certo numero di membri del comitato, ma grazie alla scelta casuale di quest'ultimo, può farlo solo se cospira con un certo numero di nodi e tra questi nodi ce ne sono, per caso, un numero sufficiente di loro selezionati come membri del comitato. Queste condizioni riducono significativamente la possibilità che un nodo manipoli il suo diritto di produrre un nuovo blocco e la probabilità di avere una fork;

2. **Meccanismo di finalità del blocco**: Esso garantisce ai blocchi (nonché alle transazioni incluse) la garanzia di sicurezza assoluta che qualifica determinati criteri. Una volta che un blocco acquisisce la sua finalità, il consenso assicurerà che non potrà essere modificato, sostituito o rimosso dalla DLT anche quando la rete incontra una situazione estremamente asincrona come quella di una fork. In teoria, un blocco può essere considerato definitivo se è confermato dal consenso BFT, che in questo caso è implementato come un processo diviso in tre fasi. In ogni fase, oltre i due terzi dei nodi devono concordare il blocco da confermare. Si può ottenere una fase del consenso BFT su un determinato blocco una volta osservato che più di due terzi dei nodi hanno partecipato alla catena che discende dal blocco. Si richiede che solo il leader e il comitato del blocco rispondano in tempo. Di conseguenza, sarà meno probabile una fork temporanea o che il servizio venga ritardato.

A questo punto è possibile spiegare perchè la PoA 2.0 è anche chiamata **SURFACE**, che sta per:

- **S**ecure: riducendo la possibilità di avere delle fork, è più difficile attaccare la blockchain, che risulta di conseguenza più sicura;
- **U**se-case adaptive: la dimensione del comitato si adatta al network e alle necessità del momento, aumentandola se è necessaria una maggiore sicurezza o diminuendola se è necessaria una maggiore velocità;
- **R**elatively **F**ork-free: uno dei problemi principali che cerca di risolvere è proprio la presenza di fork
- **A**pproach of **C**hain **E**xtension: PoA 2.0 infatti è un algoritmo di consenso per allungare la catena principale.

2.2 Transazioni

La blockchain `VeChainThor` implementa un modello di transazione migliorato per risolvere alcuni problemi comuni in altre blockchain.

```
type Transaction struct {
        body body
}

type body struct {
        ChainTag      byte
        BlockRef      uint64
        Expiration    uint32
        Clauses       []*Clause
        GasPriceCoef  uint8
        Gas           uint64
        DependsOn     *thor.Bytes32 `rlp:"nil"`
        Nonce         uint64
        Reserved      reserved
        Signature     []byte
}
```

2.2.1 Transaction Uniqueness

Per evitare possibili replay attack è necessario definire le transazioni in modo univoco tramite un transaction ID definito in questo modo:

$$TxID = hash(hash(transaction\_body\ senza\ la\ firma),\ address\_signer)$$

L'assenza della firma digitale rende possibile il trasferimento della transazione da chi la crea a chi effettivamente la esegue. A differenza di `Ethereum`,

`VeChainThor` genera `TxID` differenti per diverse transazioni dello stesso account e, pur se inviate allo stesso momento, vengono processate in maniera indipendente evitando possibili fallimenti a catena in caso una delle transazioni dovesse fallire.

2.2.2 Multi-Task Transaction (MTT)

`VeChainThor` permette ad una singola transazione di effettuare diversi task tramite l'introduzione del campo `Clause`, una struttura che rappresenta un singolo task e ne permette la concatenazione. I task sono inclusi nella singola transazione e vanno considerati atomici, cioè possono solo avvenire tutti oppure fallire tutti e vengono eseguiti nell'esatto ordine di definizione. Il meccanismo multi-task è utile per condurre processi multi-step oppure gestire casi di distribuzione fondi o registrazione in massa di prodotti

```
// struttura della transazione definita in transaction.go
type Clause struct {
        body clauseBody
}

type clauseBody struct {
        To *thor.Address `rlp:"nil"` // recipient's address
        Value *big.Int // amount transferred to the recipient;
        Data  []byte //input data
}
```

2.2.3 Forcible Transaction Dependency

`VeChainThor` fornisce un meccanismo sicuro che consente agli utenti di forzare una transazione a dipendere dal successo di un'altra. Il campo `DependsOn` salva il `TxID` della transazione da cui dipende: il sistema processa la transazione attuale solo se la transazione referenziata dal campo `DependsOn` è già presente nel libro mastro (ledger) ed è stata eseguita con successo, in quanto una transazione può finire il gas oppure venire annullata tramite uno smart contract, entrando nello stato *reverted*.

```
// check depended tx
if dep := tx.DependsOn(); dep != nil {
        found, reverted, err := findTx(*dep)
        if err != nil {
            return nil, nil, err
        }
        if !found {
                return nil, nil,
                    consensusError("tx dep broken")
        }
        if reverted {
```

```
                return nil, nil,
                    consensusError("tx dep reverted")
        }
}
```

Questo codice mostra i controlli fatti dal sistema: per prima cosa, tramite la funzione `DependsOn()` controlla se l'attuale transazione referenzia qualche `TxID` da cui dipende. In caso il puntatore non sia `NULL`, richiama la funzione `findTx(*dep)`, che si occupa di controllare lo status del `TxID`. La parte finale del codice controlla se la transazione referenziata esista e che non sia stata rifiutata; altrimenti, se uno dei due controlli fallisce, la transazione viene rifiutata.

2.2.4 Transaction Lifecycle Control

`VeChainThor` fornisce all'utente un modo per tenere traccia dello status delle transazioni inviate, utile in casi di sovraccarico della blockchain, perché permette agli sviluppatori e agli utenti di avere un certo controllo su quando verrà eseguita oppure abbandonata la transazione. Il Lifecycle Control è possibile grazie ai campi `BlockRef` e `Expiration`:

- `BlockRef` salva un riferimento ad un particolare blocco che ha come blocco successivo il primo blocco in cui la transazione corrente può essere inserita. In particolare, i primi 4 byte contengono l'altezza h del blocco mentre i secondi 4 indicano se quel blocco è già noto durante la creazione della transazione. In caso affermativo, indicano i primi 8 byte dell'ID del blocco ad altezza h;

- `Expiration` salva un numero che, sommato ai primi 4 byte di `BlockRef`, specifica l'altezza dell'ultimo blocco in cui può essere inclusa quella transazione, ovvero quando scadrà;

2.2.5 Transaction Fee Delegation

La diffusione globale delle dApps è frenata dalla necessità di conoscere e acquistare criptovalute per poter pagare le transaction fee. Transaction fee delegation è un metodo che permette a qualsiasi utente di usare le dApp senza dover acquistare criptovalute e di pagare direttamente le transaction fees, rendendo l'esperienza utente simile ad un qualsiasi sito web o applicazione mobile. Attualmente sono presenti due protocolli: the *Multi-Party Payment(MPP)* e *Designated Gas Payer(VIP191).*

Multi-Party Payment (MPP)

MPP permette, sotto certe condizioni, di far pagare le transaction fee a qualcun altro che non sia il mittente della transazione. Oltre al solito mittente e destinatario della transazione, `VeChainThor` definisce altre 3 entità:

1. *Sponsor*: account che sponsorizza il mittente e paga per lui le transaction fee;
2. *User*: `VeChainThor` dà la possibilità a qualsiasi account di registrare altri utenti come suoi utenti, permettendogli di pagare per loro le transaction fee;
3. *Credit*: `VTHO` rimanente per pagare i costi di transazione per un determinato utente di un determinato account;

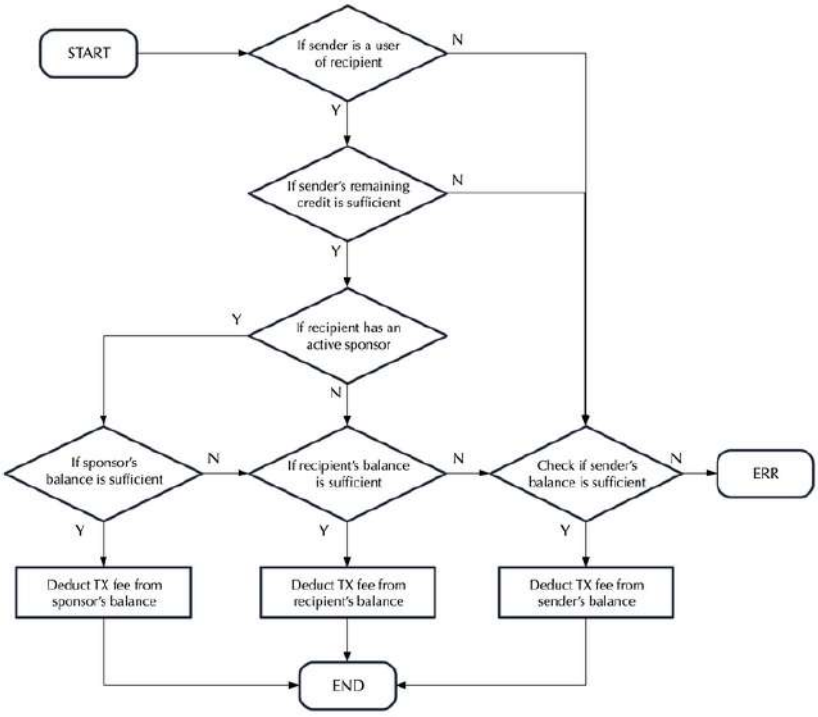


Figura 2.2: **Workflow di MPP**

`VeChainThor` per prima cosa cerca di capire se il mittente è registrato come user del ricevente e, in caso positivo e con credito sufficiente, deduce il costo della transazione dallo sponsor, se presente, oppure+ dal ricevente. In caso contrario viene dedotto dal credito del mittente, oppure la transazione è rifiutata. MPP è implementato come smart contract `Prototype` nel blocco di genesi e fornisce alcuni strumenti agli sviluppatori per poter semplificare la gestione delle loro dApp.

- `Master account` È l'account che si occupa di gestire il sistema: registra/rimuove *Users*, setta il *Credit Plan*, seleziona lo *Sponsor* attivo per l'account.

- `Credit Plan` È un metodo di protezione, messo in atto dal possessore dello smart contract, che permette di fissare un limite al pagamento delle transazioni per gli utenti.

Designated Gas Payer(VIP191)

VIP191 è una miglioria del protocollo MPP, ideato per dare più flessibilità nella delega del pagamento dei costi di transazione. MPP era stato ideato dal punto di vista dei possessori delle dApp e il protocollo funziona solo per le transazioni mandate a quei contratti. In più MPP necessita di scrivere dati sulla blockchain e quindi causa un certo overhead, perciò, non è indicato per relazioni temporanee tra utenti o utente-sponsor. VIP191 dà la possibilità al mittente di delegare il pagamento delle transaction fee ad una qualunque terza parte disposta a pagare. Per utilizzare VIP191 il mittente deve attivare la VIP-191 feature e, sia il mittente e sia chi paga le fee (il gas-payer), devono mettere la loro firma digitale nella transazione. Dopo che la transazione è accettata ed eseguita, la fee è scalata dal bilancio di chi si è firmato come gas-payer.

Implementazione di VIP-191

VIP191 ha portato due grandi cambiamenti:

1. Estensione del modello di transazione per aggiungere la firma del *Delegator*;

2. Aggiunta del campo flag del *Fee Delegated* per decidere chi sia il gas-payer in una transazione con VIP191 attivo.

2.2.6 Estensione del modello di transazione

Il campo *Reserved* nel body della struttura della transazione è stato ridefinito per essere di tipo *reserved* come mostrato:

```
type reserved struct {
        Features  Features
        Unused    [] rlp.RawValue
}
```

Il campo *Features* è un unsigned integer di 32 bit usato come bit map: ogni bit, che può essere 1 o 0, indica se una certa feature è attiva oppure no e per VIP-191 è settato a 1 il bit meno significativo. La firma del gas-payer invece è concatenata alla firma del mittente, assegnata nel campo campo *Signature*. Per evitare replay attack è necessario che il gas-payer firmi il *TxID*, che è univoco per ogni transazione.

2.3 Built-in contracts

In `VeChainThor` sono presenti 7 smart contracts preinseriti nel blocco di genesi. La loro presenza facilita l'uso della blockchain e fornisce metodi necessari a svolgere varie funzioni:

1. `Authority` fornisce metodi per interfacciarsi con gli Authority Masternodes (AMs);
2. `Energy` fornisce interfacce per le operazioni con i VTHO;
3. `Executor` fornisce metodi per facilitare la on-chain governance sulla `VeChainThor` blockchain;
4. `Extension` Fornisce metodi agli smart contracts per ottenere informazioni runtime sui blocchi precedenti;
5. `Extension-v2` Estensione di *Extension* che definisce un nuovo metodo per interrogare chi paga le fee della transazione, utile in casi di gas-payer delegato;
6. `Params` Fornisce accesso ai parametri globali quali il reward ratio e il prezzo base del gas;
7. `Prototype` fornisce l'implementazione e i metodi per gestire MPP: *setMaster*, *setCreditPlan*, *userCredit*, *addUser*, *removeUser*, *selectSponsor*;

Capitolo 3

Modello economico: two-token design

Come già accennato, è evidente che nessun'azienda costruirebbe il proprio business su una blockchain i cui token hanno un valore impredicibile. Spesso tale valore è oggetto di speculazione e rialzi o ribassi improvvisi (si pensi alla crescita del +500% di bitcoin in un anno) e quindi a rischio. Pensando a bitcoin, i costi transazionali sono espressi in BTC e quindi tali costi sono fortemente legati al valore della valuta. L'obiettivo di un modello a due token è quello di rendere meno macchinoso e lento il funzionamento del sistema (per esempio nei casi di congestione del network) e proteggere il costo del funzionamento della blockchain dalle speculazioni di mercato, slegando i costi per il funzionamento della chain da quello del token primario.

VET	**VTHO**
É il token principale ed è ciò che dà valore al VTHO, il token duale. Agisce da transfer di valore e da riserva.	Può essere pensato come il gas necessario per effettuare le transazioni relative a pagamenti ed esecuzione di smart contract.

Dal momento che il VET rappresenta il diritto ad usare la VeChain, il modello è costruito in modo che il VTHO sia generato automaticamente a partire dai VET.
Quindi chiunque abbia VET può usare la VeChain gratuitamente fino a quando l'ammontare del VTHO generato supera il consumo speso nelle transazioni eseguite. In questo modo si slega il costo delle transazioni dalla volatilità di mercato relativa al VET.

3.1 Generazione del VTHO a partire dal VET e modello di spesa

Il modello per la generazione del token duale VTHO (3.1) e per la spesa dello stesso (3.2) è il seguente:

Modello di Generazione

$$E^{gen} = v \cdot V \cdot t \tag{3.1}$$

Modello di Spesa

$$E^{cons} = p \cdot G \tag{3.2}$$

Dove:

→ E^{gen} è l'ammontare di VTHO generato

→ V è l'ammontare di VET posseduto

→ v è la velocità di generazione del VTHO dal VET

→ t rappresenta il tempo misurato in numero di blocchi inseriti.

→ E^{cons} è l'ammontare di VTHO consumato nella transazione

→ G è il "gas" richiesto per portare a termine un'operazione. VeChain prezza le operazioni in gas (in unità di mille gas)

→ p è il prezzo del gas in VTHO e gli utenti hanno libertà di modificarlo in un range prestabilito

La prima relazione evidenzia quindi che per ogni blocco inserito nella blockchain vengono generati $v \cdot V$ token di tipo VTHO (ovvero v token per ogni unità di VET posseduta). La seconda relazione mostra come il VTHO è speso nelle transazioni. Quando una transazione viene inserita in un blocco, il sistema calcola la quantità G di "gas" richiesta per quella particolare transazione e quindi E da 3.2. Il p è impostato dall'iniziatore della transazione e valori più alti di p tendono ad essere associati a maggior priorità nella processazione. Il problema dell'instabilità del prezzo sorge con l'aumento del consumo dei VTHO per le transazioni. Si tratta quindi di tarare i parametri v e p in modo da riequilibrare i prezzi.

Un esempio con i parametri attuali

Attualmente il rate v è $5 \times 10^{-8} \frac{\text{VTHO}}{\text{VET} \times \text{BLOCCO}}$. VeChain genera in media un blocco ogni 10 secondi, quindi 6 al minuto e $6 \times 60 \times 24 = 8.640$ blocchi al giorno. Per $10K$ VET verranno quindi generati in un giorno

$$E = 5 \times 10^{-8} \frac{\text{VTHO}}{\text{VET} \times \text{BLOCCO}} \times 10^3 \text{ VET} \times 8640 \text{ BLOCCHI} = 4.32\text{VTHO}.$$

Osservazione

Per evitare che p sia tale da rendere il costo della transazione troppo basso, gli utenti possono scegliere il p nell'intervallo $[p_{min}, 2p_{min}]$, dove $p_{min} = 1$ attualmente. Tali valori che limitano superiormente e inferiormente il VTHO usato per le transazioni evitano che alcuni utenti riducano la priorità di altre transazioni con fee elevate. Inoltre, in base alla domanda del mercato per il VTHO è possibile aggiustare il range per il prezzo del gas p_{min}. Se nel lungo termine anche questa scelta non stabilizza i costi di transazione (per via di un'elevata e crescente richiesta di inserimento di transazioni che porta ad accrescere il valore del VTHO), è possibile intervenire sulla generazione del VTHO modificando la velocità v. L'offerta del VTHO è basata sulla velocità corrente. La domanda di VTHO viene stimata da diversi modelli di previsione econometrici testati periodicamente rispetto a tutti i dati disponibili. Entrambi i cambiamenti all'algoritmo (per il p_{min} o il v) del modello devono essere valutati e approvati dagli stakeholder dell'ecosistema (aziende, dApp, possessori di VET, Authority Nodes) usando gli strumenti di voto decentralizzato. Raggiunta una maggioranza, la Steering Committee della Foundation provvederà al cambiamento dei parametri.

3.2 Risvolti nell'economia reale

É opportuno riassumere come un utente potrebbe quindi utilizzare la blockchain e pagare le transazioni.

- Se gli utenti (es. aziende e imprese) non intendono gestire criptovaluta e vogliono acquistare il VTHO necessario all' elaborazione delle transazioni, possono farlo direttamente dal mercato.
- Possono detenere VET, generando da sé il VTHO necessario per ogni processo operativo sulla catena secondo il modello descritto.
- Possono detenere il VET, produrre VTHO e venderlo sul mercato agli utenti interessati alla parte operativa.
- Gli utenti possono appoggiarsi a un esterno che gestisca le fee al posto suo.

Capitolo 4

La Governance

Nonostante la decentralizzazione sia il fondamento della tecnologia della blockchain, nella sua forma pura essa ha il difetto di portare ad inefficienza e all'inabilità di condurre iterazioni veloci. Per tale motivo è necessario stabilire un sistema di governance, che sia trasparente ed efficiente, in modo da permettere un'innovazione rapida e continua, e che risulti in delle decisioni che bilancino i punti di vista di tutti gli stakeholder della blockchain.

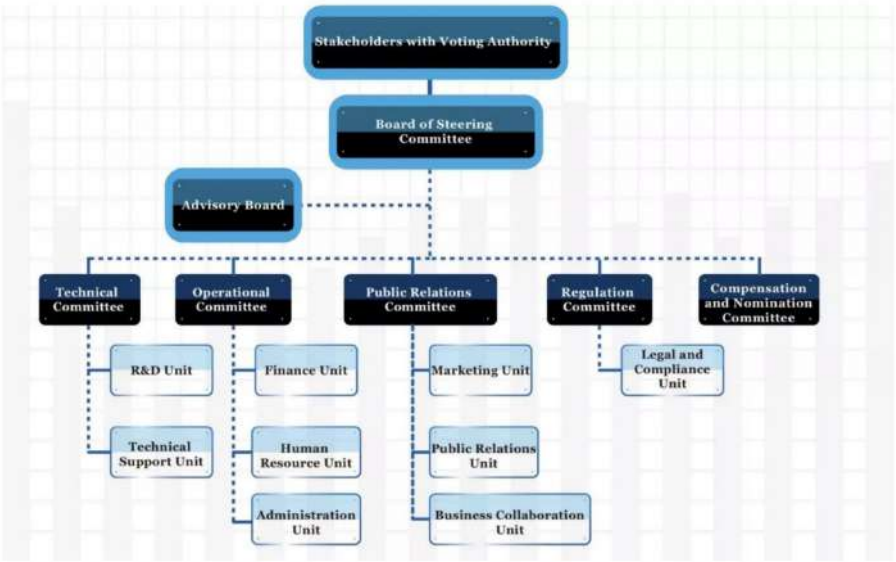


Figura 4.1: **Struttura di governance.**

4.1 La struttura di governance della Fondazione

Il meccanismo decentralizzato della tecnologia della blockchain garantisce alla Fondazione una struttura di governance unica. Il diagramma in Figura 4.1 fornisce una visione stilizzata della struttura di governance attuale.
Esaminiamo brevemente i ruoli di ogni blocco di questa struttura.

4.1.1 Stakeholders con autorità di voto

Nell'ecosistema VeChain, gli stakeholder con autorità di voto sono divisi in tre categorie, ognuna con diversa autorità di voto: gli Authority Masternodes, gli Economic X Nodes e gli Economic Nodes. Ogni categoria è, a sua volta, composta di vari livelli con autorità di voto differente. Gli stakeholders possono essere individui, enti, agenzie governative, organizzazioni no-profit e altre istituzioni interessate all'ecosistema VeChain.

Authority Masternodes (AM)

Gli Authority Masternodes sono i maintainer della blockchain e sono gli unici nodi autorizzati a validare ed inserire blocchi nella blockchain `VeChainThor`, venendo remunerati con il 30% delle commissioni di transazione in ogni blocco. Al momento ci sono 101 AM attivi, appartenenti a enti o individui la cui identità deve essere stata verificata dalla Fondazione e che sono responsabili delle loro attività e dei loro obblighi nell'ecosistema.
Un AM è un server connesso alla rete che detiene una copia completa della blockchain `VeChainThor`. Essi sono autorizzati ad inserire blocchi per mezzo dello smart-contract preinserito `Authority`. Tale contratto richiede un'autorizzazione multi-signature da parte dello Steering Committee e serve ad inserire il nodo nella whitelist che autorizza a modificare la blockchain.
Come ricompensa per mantenere l'integrità della blockchain, contribuire all'ecosistema VeChain e partecipare ai meccanismi di governance, la rete retribuisce gli AM con il 30% dei token VTHO pagati come commissioni di transazione della blockchain, mentre il restante 70% viene bruciato. Essendo la blockchain `VeChainThor`, come detto in precedenza, basata sulla PoA, gli AM non competono per validare i nuovi blocchi, ma vengono piuttosto selezionati casualmente per farlo.
I possessori di Authority Masternodes possono appartenere alle seguenti categorie (ma non limitatamente ad esse):

- Utenti di imprese
- Team di sviluppo di blockchain
- Partner di sviluppo tecnico e commerciale
- Collaboratori della community
- Partner di ricerca

La loro identità viene verificata dalla Fondazione ma, vista la preferenza di alcuni possessori di AM (in particolar modo degli utenti provenienti dalle imprese) di mantenere i loro dati e le loro attività privati nei confronti del pubblico, la Fondazione ha deciso che la scelta di svelare il proprio stato come detentore di un Authority Masternode o meno è a discrezione del possessore stesso. Ciononostante, i candidati a diventare AM saranno favoriti nella selezione in caso siano disponibili a rendere pubblico il loro stato.
Infine, le loro performance vengono costantemente misurate dal team operativo della Fondazione e il non rispetto di determinati standard può condurre alla loro squalifica in qualità di AM.

Economic X Nodes (XN)

Gli Economic X Nodes, insieme agli Economic Nodes, furono creati durante le prime fasi dell'ecosistema per iniziativa della Fondazione. Gli XN possono appartenere a quattro livelli:

- Mjolnir X Node (MX)
- Thunder X Node (TX)
- Strength X Node (SX)
- VeThor X Node (VX)

Secondo le regole stabilite dalla Fondazione, nessun nuovo XN può essere creato, ma essi possono solo aumentare il loro livello. Di conseguenza il numero di Economic X Nodes andrà a decrescere col tempo.

Economic Nodes (EN)

Anche gli Economic Nodes, come detto, furono creati per iniziativa della Fondazione, ma, a differenza degli XN, chiunque possieda dei VET può fare domanda per diventare un EN. Di conseguenza, il numero di questi ultimi può crescere. La domanda può essere presentata per mezzo di uno smart contract da qualsiasi indirizzo che possieda il numero minimo di VET necessario (specificato nella tabella sottostante). Gli EN possono appartenere a tre livelli:

- Mjolnir Node (M)
- Thunder Node (T)
- Strength Node (S)

Gli EN e gli XN non validano transazioni sulla blockchain, ma offrono stabilità all'ecosistema e forniscono vantaggi rispetto agli utenti di VeChain che non possiedono nodi.

Riassumiamo nella tabella sottostante la suddivisione in categorie e livelli dei nodi della blockchain. Per ogni livello è specificato il numero minimo di VET da possedere e il periodo minimo di maturità per appartenerà alla categoria corrispondente (si noti che il periodo di maturità dei nodi VX non è disponibile in quanto, come precedentemente specificato, non possono essere creati nuovi nodi XN). È inoltre indicato il numero di voti che corrispondono ad ogni nodo in caso di votazioni e all'autorità di voto delle categorie.

Categoria	Livello del nodo	Possesso minimo di VET	Periodo di maturità	Voti per nodo	Autorità di voto
Authority Masternodes (AM)	N/A	25.000.000	N/A	1 voto AM	**40%**
Economic X Nodes (XN)	MX	15.600.000	90 giorni	26 voti XN	**40%**
	TX	5.600.000	60 giorni	10 voti XN	
	SX	1.600.000	30 giorni	3 voti XN	
	VX	600.000	N/A	1 voto XN	
Economic Nodes (EN)	M	15.000.000	30 giorni	15 voti EN	**20%**
	T	5.000.000	20 giorni	5 voti EN	
	S	1.000.000	10 giorni	1 voto EN	

Tabella 4.1: Suddivisione in categorie e livelli dei nodi della blockchain

4.1.2 La Commissione del Comitato di Governo (Board of Steering Committee)

Lo Steering Committee è il corpo di governo della Fondazione VeChain e ha il compito di definire le strategie e selezionare le poltrone dei Comitati Funzionali. Le funzioni pricipali della Commissione sono:

1. Proporre e organizzare le votazioni su problematiche relative alla blockchain;

2. Controllare, approvare e monitorare le attività strategiche, tecniche e finanziarie della Fondazione, oltre al suo stato finanziario, inclusi la proprietà di VET e l'impiego degli utili;

3. Controllare, modificare e approvare i principi di governance della Fondazione;
4. Controllare, approvare e monitorare le procedure di nomina e elezione dei membri dello Steering Committee, dei Comitati Funzionali e del Segretario Generale della Fondazione;
5. Autorizzare o revocare i validatori del consenso (ovvero gli Authority Masternodes);
6. Controllare, approvare e monitorare il modello operativo dei VTHO, il modello di valutazione dei VET e la velocità di generazione dei VTHO a partire dai VET o qualsiasi altra attività on-chain incarnata da uno smart contract distribuito su `VeChainThor`.

La Commissione è eletta dagli stakeholder con autorità di voto ogni due anni per mezzo di una votazione presieduta da un comitato apposito (il Comitato delle Nomine). É composta da rappresentanti della Fondazione VeChain, da detentori di Authority Masternodes, da sviluppatori, utenti di imprese, partner commerciali e membri indipendenti.

4.1.3 Comitato di Consulenza (Advisory Board)

La Fondazione nomina un Comitato di Consulenza che affianchi lo Steering Committee per mezzo del contributo di membri con un ampio spettro di competenze che possano fornire suggerimenti sul mondo dell'industria.

4.1.4 Comitati Funzionali (Functional Committees)

Lo Steering Committee ha inoltre istituito i seguenti Comitati Funzionali: Tecnico, Operativo, Pubbliche Relazioni, di Regolamentazione, di Retribuzione e delle Nomine. Ogni comitato è presieduto da un membro dello Steering Committee o dell'Advisory Board e include membri con esperienza nel corrispondente campo.

4.2 Meccanismo di on-chain governance

La On-Chain governance di `VeChainThor` è basata sulle decicisione prese dagli stakeholders o dall'organo di governo. La on-chain governance consiste in 3 fasi: decision making, authorization and execution.

Figura 4.2: **On-chain governance.**

- `Decision making` É la prima fase, in cui si prendono le decisioni su cosa eseguire sulla catena. Le decisioni sono ottenute tramite voto sia condotto tramite un contratto di voto oppure off-chain all'interno del governing body. Il primo metodo fornisce massima trasparenza e spesso coinvolge tutti gli stakeholders, mentre il secondo è più efficiente e agile da applicare.
- `Authorization` É la seconda fase, dove una proposta votata è passata al governing body per l'approvazione finale. Ogni proposta deve essere approvata da una maggioranza di membri del governing body come misura di sicurezza extra per evitare eventuali attività malevole sulla catena.
- `Execution` È la fase finale della on-chain governance: dopo che una proposta è stata approvata della maggioranza, chiunque può far partire l'esecuzione dell'azione sulla catena definita dalla proposta.

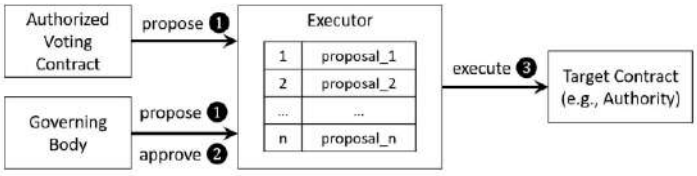


Figura 4.3: **Framework for on-chain governance.**

4.2.1 Implementazione

La `VeChainThor` blockchain fornisce un framework flessibile per implementare la on-chain governance tramite lo smart contract *Executor*.

Fase Authorization

Executor fornisce i metodi *propose* e *approve* per eseguire la fase di autorizzazione. I membri del governing body oppure utenti autorizzati possono invocare la seguente funzione per inviare una proposta:

```
propose(target_contract_address, encoded_data)
```

Una proposta è un'istanza della struttura *proposal* in *Executor* ed è creata dalla funzione *propose*. Alla sua creazione è generato un *propsalID* univoco. Dopo che la proposta è salvata in *Executor*, i membri del governing body hanno una settimana per autorizzarla tramite l'invocazione della funzione *approve*.

Fase Execution

Se la proposta viene accettata dalla maggioranza richiesta, chiunque può invocare la funzione *execute* per far partire l'esecuzione dell'azione sulla catena tramite la funzione di basso livello:

```
target_contract_address.call(encoded_data)
```

4.2.2 Voto di tutti gli stakeholder

Prima di una votazione di tutti gli stakeholder della blockchain, la Fondazione annuncia le regole dettagliate (giorno, periodo di voto e percentuale minima di partecipanti per ogni categoria). Alla votazione possono partecipare tutti i nodi con stato attivo e, affinché essa possa essere ritenuta valida, il numero di partecipanti per ogni categoria deve superare una soglia fissata. In caso contrario l'autorità di voto viene modificata in accordo al numero di votanti per ogni categoria, in modo da evitare manipolazioni da un numero ridotto di nodi.

Aggregazione dei voti

Una volta terminata una votazione, il risultato finale può essere calcolato come

$$V = \omega_{AM} V_{AM} \times \omega_{XN} V_{XN} \times \omega_{EN} V_{EN},$$

dove V_{AM}, V_{XN} e V_{EN} sono i risulati di voto delle singole categorie e ω_{AM}, ω_{XN} e ω_{EN} (con $\omega_{AM} + \omega_{XN} + \omega_{EN} = 1$) sono i loro rispettivi pesi. I pesi sono quelli specificati dalla tabella precedente, ma essi possono essere modificati dallo Steering Committee quando ritenuto appropriato.

4.3 Gestione finanziaria

La gestione finanziaria della Fondazione è affidata a un team di gestione finanziaria, il quale ha il compito di gestire la pianificazione e il controllo finanziario, la contabilità e la conformità legale dei progetti per coadiuvare l'amministrazione nelle fasi decisionali.

4.3.1 Fonti di finanziamento

Essendo un organizzazione no-profit, la VeChain Foundation non distribuisce profitti o dividendi al team fondatore o agli azionisti della Fondazione. Ciononostante, la Fondazione cerca attivamente fonti di guadagno per rendere sostenibile il loro progetto e sviluppare l'ecosistema VeChain. Vi sono vari canali di profitto:

- **Gestione degli asset e investimenti**: il 10% del capitale è investito in ulteriori progetti innovativi che possano creare valore per l'ecosistema; inoltre vengono eseguiti degli investimenti in monete fiat e criptovalute (come BTC e ETH) per mitigare il rischio di volatilità del mercato;
- **Servizi professionali**: la Fondazione a volte riceve dei pagamenti in monete fiat o asset digitali per i servizi forniti (come fornire consulenza ad imprese che vogliano integrare la tecnologia della blockchain nel loro business);
- **Servizi con supporto ai VTHO**: nel lungo termine, una volta raggiunto un equilibrio del modello economico della VeChain, la Fondazione riceverà dei proventi dai VTHO generati dalla sua riserva di VET.

Capitolo 5

Utilizzi della VeChain

Vengono presentate in questa sezione alcune delle principali applicazioni della catena VeChain che si allontanano da ambiti più battuti come il mercato della valuta e il settore della finanza.

5.1 Provenienza di cibi e bevande

5.1.1 Il problema

In un mondo estremamente globalizzato e i cui gli scambi commerciali raggiungono livelli sempre più alti, non sono infrequenti scandali alimentari, contraffazione di cibi e bevande di marchio protetto e falsificazione di informazioni circa la provenienza degli alimenti. I consumatori diventano sempre più attenti alla provenienza degli alimenti e si chiedono se ciò che riporta l'etichetta sia affidabile o frutto di contraffazione della lunga catena di produzione che porta dal produttore al supermercato. Si ricordano alcuni avvenimenti che hanno destato particolare clamore negli ultimi 15 anni:

- Latte contaminato con melammina in Cina che ha portato a un grave quadro clinico centinaia di bambini e ha causato la morte di alcuni di essi [2].
- Pollame scaduto da quarant'anni più volte congelato, scongelato e trattato per prevenirne l'imputridimento [9].
- Maiale contaminato dal clenbuterolo, sostanza che accelera lo sviluppo dei maiali ma è tossica per l'organismo umano: resa illegale dal 2002, è ancora usata da alcuni allevatori fraudolentemente [1].

5.1.2 La soluzione

VeChain permette ai partecipanti della catena di produzione di un qualsiasi bene alimentare di collaborare su una piattaforma di dati visibile a tutti. Que-

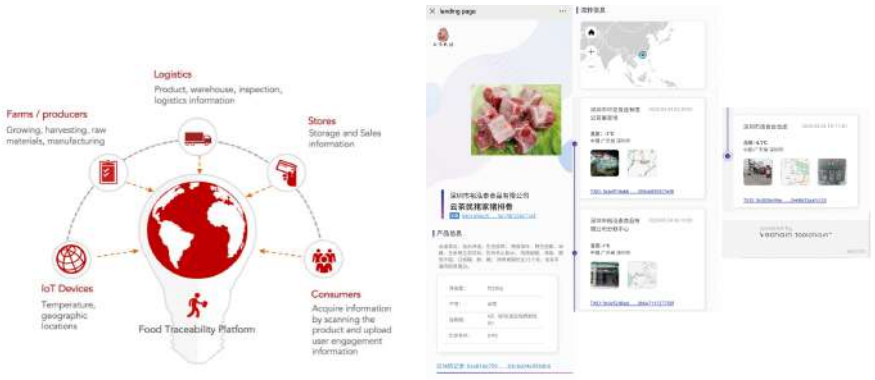


Figura 5.1: In alto: le varie tappe della catena di produzione. In basso: Esempio di informazione accessibile tramite il QR code di un alimento in un supermercato. La scansione dà accesso a tutti i blocchi legati all'alimento

sta è infatti fruibile tramite la scansione di un QR code posto sulle confezioni alimentari. Su questa piattaforma sono visibili:

1. Origine degli ingredienti del prodotto
2. Tracciamento di ciascuna località in cui il prodotto è stato trasportato
3. Report di ispezioni e controlli di temperatura e altre informazioni

Queste informazioni sono inserite da coloro che producono i dati e firmate crittograficamente.

Tutti i gestori della catena di distribuzione hanno interesse a firmare e inserire il loro contributo, in modo che eventuali enti con intento di frode non dispongano di queste stesse informazioni sulla piattaforma digitale legata ad un alimento. Per i contributori alla VeChain sono inoltre disponibili sensori che rilevino temperatura, umidità o accelerazione dell'ambiente in cui si trova un alimento: il consumatore è a conoscenza di tutto.

La portata dell'introduzione della blockchain per garantire la provenienza e la filiera logistica del cibo sono molto ampie: ogni alimento è collegato alla sua storia che risulta immodificabile (a breve si vedrà come evitare il problema della duplicazione dei QR code).

5.1.3 Vantaggi del cambiamento

- La fiducia del consumatore aumenta e quest'ultimo tende ad indirizzare il suo interesse verso marchi che offrono questo servizio
- La reputazione delle grosse industrie aumenta
- Qualità alimentare promossa e miglioramento anche del controllo della catena di produzione

5.1.4 Caso reale: Walmart

Nel giugno 2019, la grossa catena Walmart China ha lanciato la "Walmart China Blockchain traceability Platform" sulla piattaforma ToolChainTM. Scannerizzando i QR code presenti sugli alimenti, i clienti raggiungono tutte le informazioni rilevanti sul prodotto.

5.2 Contraffazione di beni di lusso

È in crescita anche il settore della contraffazione, con traffici che arrivavano a 1200 miliardi di dollari nel 2017 e che ora raggiungono 1820 miliardi di dollari a livello globale. Ha inoltre preso sempre più terreno la rivendita di beni di lusso, mercato poco controllato e anche il commercio online è ormai un colosso di proporzioni inimmaginabili.

5.2.1 Il problema

Soprattutto con il commercio online, è difficile riconoscere un bene (alimentare o non) che sia di qualità e non abbia provenienza fraudolenta. Vini creati con polveri, vestiti di scarsa qualità e colorati con tinte dannose alla salute, scarpe o giacche contraffatte sono solo alcuni dei beni oggetto di frodi frequenti.

5.2.2 La soluzione

La VeChain permette ai marchi grandi e piccoli di legare il prodotto fisico al mondo digitale attraverso dei circuiti NFC. Questo meccanismo supera il codice QR e lo potenzia, non consentendo la duplicazione del QR e impossibilitando la rimozione di un circuito NFC da una confezione originale. In qualche modo tale tecnologia aiuta a rendere inviolabile l'integrità fisica oltre che digitale di un prodotto. Le informazioni riportate sono quelle rese disponibili sulla ToolChainTM.

5.3 Ecosistema digitale per basse emissioni di anidride carbonica

É ormai dagli anni 60 che le Nazioni Unite e le società più tecnologicamente avanzate, seguono il problema dell'inquinamento ambientale e del surriscaldamento globale.

5.3.1 Il problema

Il problema è riuscire a misurare quantitativamente quanto il produttore si impegni alla sostenibilità e a ridurre le emissioni. Non si riescono quindi a certificare in maniera corretta le emissioni, le polizze di quantificazione non sono trasparenti e probabilmente i risultati sono quantomeno alterabili.

5.3.2 La soluzione

VeChain e DNV GL hanno inzializzato un ecosistema di basse emissioni con alcune compagnie che faranno da apripista per il progetto. In questo ecosistema a basso impatto i ruoli sono:

- CER (carbon emission reduction) generator. La piattaforma digitalizza, qualifica e quantifica i comportamenti degli utenti come contributori al basso impatto ambientale. Questa quantificazione oggettiva è condotta con supporto dell' IoT, uso di stazioni di ricarica elettrica, percentuale di uso di mezzi elettrici, uso del trasporto pubblico.
- CER validator. Gli esperti di DNV GL verificano il comportamento degli utenti in base ai dati raccolti e il processo di validazione è eseguito tramite smart contracts per maggior trasparenza.
- CER consumer. Sponsor come imprese e le organizzazioni che intendono fornire prodotti e servizi per motivare i singoli collaboratori di CER.
- Sostenitore Tecnico. Lancia l'infrastruttura tecnica e e fornisce supporto per mantenere la piattaforma.

Bibliografia

[1] AsiaNews.it. *Nell'Henan, carne di maiale contenente "veleno". Sospesi i responsabili sanitari locali*. http://www.asianews.it/notizie-it/Nell27Henan,-carne-di-maiale-contenente-E2809CvelenoE2809D.-Sospesi-i-responsabili-sanitari-locali-21054.html. 2011.

[2] Fabio Cavalera Corriere.it. *Scandalo del latte al veleno in Cina Famiglie in rivolta, consumatori in marcia*. https://www.corriere.it/esteri/08_settembre_21/21_CRONACHE_LATTE_bd3ab0aa-87af-11dd-b5e4-00144f02aabc.shtml?refresh_ce-cp. 2008.

[3] VeChain Foundation. *Blockchain Governance: How Decisions Are Made on VeChainThor*. https://www.youtube.com/watch?v=Vl6Ssr-1Nq0. 2020.

[4] VeChain Foundation. *VeChain Foundation Docs*. https://docs.vechain.org/thor/learn/fee-delegation.html/.

[5] VeChain Foundation. *VeChain Foundation Docs*. https://docs.vechain.org/thor/learn/transaction-model.html/.

[6] VeChain Foundation. *VeChain Foundation Whitepaper*. https://www.vechain.org/whitepaper/.

[7] VeChain Foundation. *VeChain's Tech Deep Dive Series - Session 2, Episode 1: Introduction to MPP and Its Implementation*. https://www.youtube.com/watch?v=2kjeZF9JbQ0/.

[8] VeChain Foundation. *What you might not know about PoA*. https://www.youtube.com/watch?v=1I3xbBJwkR4&t=3136s. 2020.

[9] Scatti di Gusto. *Sequestrata carne vecchia di 40 anni pronta per il mercato cinese*. https://www.scattidigusto.it/2015/06/29/sequestro-carne-avariata-cina/. 2015.

[10] VeChain Italia. *VET e VTHO: il modello economico a due token di VeChain*. https://vechainitalia.net/2019/11/16/vet-e-vtho-il-modello-economico-a-due-token-di-vechain/. 2020.

[11] Brot KnoblauchHaus. *Consensus mechanisms in Blockchains for the layperson*. https://medium.com/the-capital/consensus-mechanisms-in-blockchains-for-the-lay-person-c6eacaa7945d. 2021.

[12] Brot KnoblauchHaus. *Crossing the chasm: How VeChain could unlock the secrets for mass adoption of Blockchain*. `https://medium.com/the-capital/crossing-the-chasm-how-vechain-could-unlock-the-secrets-for-mass-adoption-of-blockchain-1ad54dda3bec`. 2021.

[13] Brot KnoblauchHaus. *VeChainThor Primer: VeChain vs private chains*. `https://bredgarlichouse.medium.com/vechainthor-primer-vechain-vs-private-chains-6dac890aad5`. 2021.

[14] Ziheng Zhou Zhijie Ren. "SURFACE: A Practical Blockchain Consensus Algorithm for Real-World Networks". In: (2020).

STEEM

Giulia Corrente, Lorenzo De Siena, Jacopo Dominici, Rachid El Amrani

Indice

Capitolo 1

Introduzione

Dando uno sguardo al sito CoinMarketCap, STEEM è quotato a circa $0,51 il 29 maggio 2021. Prima però dobbiamo porci la grande domanda: che cos'è Steem?
Steem è stato costruito per dare voce alle persone in tutto il mondo. Steem è una piattaforma social blockchain fondata nel 2016. Diversamente da altre criptovalute, si pone l'obiettivo di remunerare tutti gli utenti che attivamente portano valore alla piattaforma.
La tecnologia che alimenta Steem ha dimostrato pubblicamente la sua capacità di elaborare migliaia di transazioni al secondo, una capacità più che sufficiente per gestire più volte l'attività di alcune popolari piattaforme di social media come reddit.com. Nelle applicazioni dei social media Steem premia gli utenti con token (Steem Dollars e Steem Power) per la pubblicazione e la cura di contenuti di qualità. Una delle applicazioni più utilizzate è steemit.com, il front-end user-friendly di Steem. Gli utenti possono creare post, ai quali viene assegnato un valore in base alla quantità di valore percepito.
Gli utenti di Steem sono incoraggiati a votare i contenuti che ritengono utili dando ad essi del peso mediatico. Maggiore è il numero di persone raggiunte con contenuti di valore, maggiore sarà la quantità di interazioni positive con esso, maggiore sarà la ricompensa guadagnata dal creatore del post e dagli utenti che lo curano (curators).
Gli utenti che detengono più valuta possono esprimere voti con più influenza rispetto a quelli che ne hanno meno. Chiunque può vendere i suoi STEEM (la moneta della blockchain) o investirli per aumentare il suo potere di voto. Le transazioni Steem possono essere completate in circa tre secondi senza alcuna commissione.
Il progetto ha in seguito integrato altre piattaforme social decentralizzate come DTube, un sito di streaming video, oppure Utopian, una piattaforma di crowdsourcing che finanzia progetti open source.
STEEM ha una offerta circolante di circa 380 milioni di monete e la circolazione aumento ogni anno con un'inflazione decrescente dello 0.5% all'anno, ed è circa il 7% quest'anno. Gli utenti possono scambiare STEEM su una varietà di

piattaforme di trading, tra cui Huobi Global e Binance.

Capitolo 2

Proof of Brain

A differenza di altre blockchain dove gli utenti guadagnano attraverso prove di lavoro oggettive come ad esempio la Proof of Work (PoW), la Proof of Stake (PoS), in cui dimostrano alla rete il loro interessamento al progetto attraverso il loro capitale investito, Steem premia anche il tempo che l'utente investe nella piattaforma per esprimere propri, soggettivi, giudizi, creare nuovi contenuti e condividere ciò che reputa più opportuno con il resto della rete.
Questo nuovo concetto di remunerazione va ad incidere direttamente sullo sviluppo della rete stessa, dando origine a quella che gli autori chiamano Proof of Brain (PoB).
Ci sono due principali modi per guadagnare tramite PoB:

- Ricompense per gli autori: quando l'utente, o un gruppo di utenti, pubblica un post o un commento se votato da altri utenti, fanno sì che gli autori guadagnino una ricompensa;
- Ricompense per la cura: quando voti un post o un commento che ritieni abbastanza utile per la piattaforma, vieni ricompensato con una quota proporzionale al peso del tuo voto come premio per averla curata.

A differenza dei social media centralizzati in cui un numero ristretto di moderatori si occupa di moderare i contenuti, qui tutti gli utenti della piattaforma sono invitati e incentivati a svolgere questo ruolo facendo sì che il risultato ultimo sia interamente frutto del volere di tutta la comunità.
Il meccanismo PoB è indirizzato dalle linee guida della comunità in cui viene specificato quale tipo di contenuto sia considerato positivo per uno specifico tag e quale invece sia da votare negativamente in modo che lo spamming, il plagio o qualsiasi altra forma dannosa per la comunità possano essere controllati.
Tutti i post, i commenti e i voti sono tradotti in transazioni della blockchain di Steem: il tutto viene archiviato permanentemente nei blocchi in cui ogni transazione viene inserita. Ciò implica che qualsiasi forma di pensiero, espressione e giudizio gode della proprietà che ogni comunità che fonda le proprie basi sui principi di libertà auspicherebbe: la resistenza alla censura.

2.1 Voting Power

Il valore di un Upvote o di un Downvote dipende dal Voting Power. Se un utente non ha mai votato prima d'ora avrà il 100% della Voting Power rimanente.
Quando si dà un voto, l'impatto che questo avrà sulla ricompensa del post o commento in cui esso è stato applicato dipenderà dalla percentuale di Voting Power (VP) che l'utente ha al momento del voto, dagli SP accumulati e da che percentuale di influenza (weight) ha applicato al momento del voto. In particolare, il compenso è proporzionale a: weight*VP*SP.
A seconda della percentuale di influenza utilizzata, il VP si ridurrà del (2*weight)%. Di default l'influenza del voto è del 100% quindi il compenso associato sarà proporzionale a VP*SP e farà diminuire la percentuale di VP rimanente del 2%.
Dato che ogni voto usa una percentuale della potenza di voto, i voti successivi (se voto per più post) varranno di meno e ci vorrà più tempo a raggiungere di nuovo la piena potenza di voto, infatti abbiamo che ogni giorno la potenza di voto si ricarica del 20% del totale. La blockchain utilizza questo metodo in modo che gli utenti con maggiore potenza di voto non prendano il comando e influenzino troppo la rete con i contenuti di loro piacimento. Inoltre, tramite questo meccanismo anche il tentativo di dividere i token in più account per avere più voti non è funzionale da parte degli utenti che vogliono il dominio della rete perché diminuisce di conseguenza la potenza di voto.

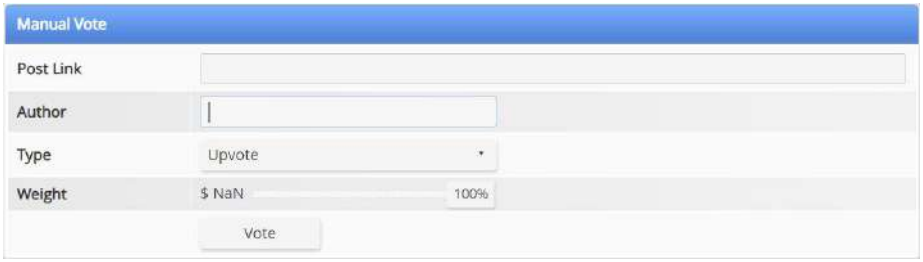


Figura 2.1: Nell'immagine si può notare come la percentuale di potenza di voto sia uno dei parametri specificabili al momento del voto.

Capitolo 3

Token di Steem

Su Steem coesistono 3 monete fortemente correlate: STEEM, Steem Power e Steem Dollars.

3.1 STEEM

Gli STEEM permettono il funzionamento dell'intero sistema Steem. Hanno un loro valore determinato dal mercato, e sono altamente volatili (quindi fortemente soggetti alle fluttuazioni di mercato), quindi non è destinato ad essere usato a fini commerciali (per acquisti di beni o servizi) ma ad essere scambiato sul mercato con altre monete a fini speculativi.
Gli STEEM possono essere scambiati infatti, sia sul mercato interno o sui vari exchange come blocktrades, bittrex, ecc.

3.2 Steem Power (SP)

Se STEEM rappresenta la moneta alla base della blockchain, liquida e scambiabile in qualsiasi momento, Steem Power (SP) rappresenta un tentativo di creare una forma di investimento nella community.
L'idea è che se qualcuno crede nella piattaforma, questo ci investirà capitale e/o tempo affinché la piattaforma acquisisca valore e, per questa ragione, la piattaforma lo ricompenserà proporzionalmente al suo investimento.
L'investimento in SP è fortemente incentivato dalla piattaforma che, per il principio su cui si basa la PoS, si fida di chi più ha investito, cioè chi di più ha da perdere da un eventuale collasso del sistema e quindi ne conferisce premi e poteri in misura proporzionale alla SP accumulata.
Il primo modo per acquisire SP è la partecipazione attiva, soggettiva (PoB), alla vita della comunità, in particolare vengono ricompensati, se gli utenti lo desiderano, per ogni attività che la piattaforma considera utile ad aumentarne il valore della comunità stessa.
L'utente riceverà una certa quantità di SP se:

- Riceve up-vote in un post/commento da lui creato
- se un utente mette un up-vote in un post/commento

La quantità di SP è proporzionale:

- a quanto il post sia considerato importante per la piattaforma (quantità giudizi positivi che il post possiede);
- a quanta percentuale di potenza di voto ogni utente che ha inserito up-vote /down-vote [1] ha utilizzato;
- a quanta SP possiede ogni utente che ha inserito up-vote /down-vote.

Un altro modo per generare SP è convertendoli a partire da STEEM, l'atto prende il nome di Powering Up.
Affinché gli SP siano uno strumento di investimento a lungo termine nel progetto, gli SP sono concepiti con l'idea di non essere scambiabili e, l'azione di riconversione degli SP a STEEM (Powering Down), viene regolata in modo da non permetterne speculazioni sul valore della moneta. In particolare, i possessori di SP riceveranno completamente gli STEEM dopo 13 settimane dall'inizio della richiesta di conversione con un rateo settimanale di 1/13 dell'ammontare spettante.
Come ulteriore incentivo, i titolari di SP guadagnano anche nuovi token in funzione alla quantità di SP che rimane in loro possesso.
In particolare, titolari di SP vengono pagati il 15% dell'inflazione annuale. La quantità di nuovi token (SP) che ricevono è direttamente proporzionale alla quantità di SP che detengono rispetto alla quantità totale di SP posseduti da tutti gli utenti.
Come vedremo la quantità di SP influenza direttamente la votazione dei Witnesses, entità fidata della blockchain a cui ne affida i compiti di stimare i tassi di cambio e a cui si delega l'emissione dei blocchi e quindi l'orchestrazione del sistema (DPoS).

3.3 Steem Dollars (SBD)

Gli SBD sono stati ideati per portare stabilità nel mondo delle criptovalute e agli individui che utilizzano la rete Steem. In particolare, l'utente che possiede 1 SBD possiede l'equivalente volume di STEEM corrispondente a $1, in questo modo la rete Steem riesce a inglobare al suo interno anche gli utenti che diffidano della volatilità delle criptovalute.
Gli SBD possono essere visti come uno strumento di debito a breve termine che in qualsiasi momento possono essere convertiti in una quantità di STEEM pari a 1$. Per assicurare che la conversione degli SBD sia ancorata al valore del

[1] L'impatto dei down-vote è il medesimo degli up-vote ma con segno negativo, cioè verranno sottratti al totale dovuto agli up-vote.

dollaro, sono eletti dai possessori di Steem Power i Witnesses che forniscono i feed dei prezzi di mercato.
Per evitare attacchi interni da parte dei Witnesses abbiamo che:

- il feed utilizzato dalla rete viene calcolato come la media dei feed dei singoli Witnesses;
- i Witnesses vengono pagati in SP, così facendo chi produce falsi feed ha solo da rimetterci.

Ciò però non risolve il problema della manipolazione del mercato a breve termine. Per evitare questo problema la piattaforma Steem oltre a prendere il feed medio dei vari Witnesses prende il feed mediano su un periodo di tre giorni e mezzo, aggiornandolo ogni ora. Così facendo se la corruzione sui feed dei prezzi dura meno della metà del periodo di tre giorni e mezzo avrà un impatto minimo sul prezzo di conversione. Inoltre se il feed viene corrotto, i partecipanti della rete hanno l'opportunità di espellere il produttore di feed corrotto prima che possa influenzare il prezzo di conversione. Infine, ogni produttore di feed ha l'opportunità di trovare e correggere i problemi prima che influenzino il prezzo di conversione. Quindi in generale i partecipanti della rete grazie alla finestra temporale dei 3 giorni e mezzo hanno circa un giorno e mezzo per trovare e risolvere i problemi.
Data la relazione tra la moneta STEEM e SBD si potrebbe verificare il Timing Attacks, ovvero se si verifica un aumento improvviso del valore di STEEM i trader potrebbero richiedere la conversione di SBD al vecchio prezzo più basso (grazie al fatto che la blockchain aggiorna il feed su una finestra temporale di tre giorni e mezzo) e vendere gli STEEM ottenuti al nuovo prezzo più alto. Per contrastare questo attacco Steem richiede che tutte le conversioni da SBD a STEEM avvengano in tre giorni e mezzo.
Il protocollo consente agli utenti di poter fare in autonomia unicamente le conversioni da SBD a STEEM, mentre per ottenere gli SBD, oltre ad ottenerli tramite Exchange, gli utenti devono partecipare all' interno della piattaforma come "creatori" o "curatori".
Ciò è giustificato dal fatto che un'erogazione incontrollata di SBD può portare a problemi di inflazione per la criptovaluta STEEM. Infatti, se il rapporto tra debito (SBD Totali) e proprietà (STEEM e SP Totali) è troppo alto allora nel momento della conversione del debito, ovvero conversioni da SBD a STEEM, vi è un aumento dell'offerta della criptovaluta STEEM e una conseguente inflazione.
In pratica se il debito diventa maggiore del 10% della capitalizzazione del mercato totale di STEEM allora la Blockchain in automatico ridurrà la quantità di STEEM generato dalla conversione di SBD al massimo pari al 10% della capitalizzazione del mercato totale di STEEM. Quindi anche non agendo direttamente sui valori di mercato di SBD e STEEM, limitando il numero di conversioni si limita dall' alto il valore del rapporto.
Oltre alla blockchain anche i Witnesses che si occupano di fornire un giusto feed del prezzo di mercato, si occupano indirettamente di controllare che il rapporto

tra debito e proprietà non diventi troppo alto. Infatti, se abbiamo un rapporto alto e 1 SBD si scambia per meno di $1 allora i Witnesses regolano i feed del prezzo di mercato verso l'alto in modo che si fornisca più STEEM per SBD, così facendo aumenta la domanda di SBD e quindi ritroviamo un rapporto uno a uno tra dollaro e SBD e il rapporto tra debito e proprietà si riduce.
Da ciò possiamo vedere come non si ha una creazione di STEEM costante ma essa si basa sull'offerta e sulla quantità di SP e SBD in circolazione.

Capitolo 4

User keys

Ogni utente in fase di registrazione è tenuto a generare 4 chiavi che gli permetteranno di autorizzare le varie operazioni nella piattaforma. Queste in ordine di importanza sono:

- Owner Key: per cambiare password, modificare le chiavi dell'account (compresa l'Owner Key stessa) e recuperare l'account;
- Active Key: per trasferire fondi, effettuare Power Up/Down, votare per Witnesses/proposte;
- Posting Key: per creare post, commentare, votare, modificare il profilo;
- Memo Key: per inviare/leggere messaggi cifrati nei trasferimenti.

In aggiunta, quando un utente vuole candidarsi ad essere uno Witness, deve scegliere una chiave che ne rappresenti la sua identità e che utilizzerà per le sue mansioni da Witness, questa chiave è chiamata Block Signing Key.

Capitolo 5

Delegated Proof Of Stake (DPoS)

In Steem viene utilizzato il protocollo di consenso DPoS, ora vediamo nel dettaglio come funziona. Gli individui che hanno investito in Steem votano per selezionare individui, chiamati Witnesses, adatti alla produzione dei blocchi. Il voto da parte degli utenti è ponderato in base all' investimento dell'individuo ovvero alla quantità di SP posseduti dall' individuo. La produzione dei blocchi avviene in round, per ogni round sono selezionati 21 Witnesses per creare e firmare blocchi di transazioni, ad esempio transazioni finanziarie e/o transazioni riguardanti creazione, commenti o votazioni dei post. In particolare, abbiamo che in ogni round ogni Witness inserisce un blocco nella blockchain dopo che questo è stato accettato da almeno 14 Witnesses, ricevendo il relativo compenso in SP. In particolare, la remunerazione dei Witnesses annuale corrisponde al 10% della quantità di STEEM erogati annualmente.
I 21 Witnesses sono scelti tramite votazioni da parte degli utenti della rete, ovvero si ha una graduatoria di tutti i Witnesses e a ogni round i primi 20 della graduatoria sono scelti come Witnesses e l'ultimo viene scelto casualmente ma con probabilità che dipende dalla posizione in graduatoria, ovvero più si è in alto nella graduatoria maggiore è la probabilità di essere scelti. Si parla di votazioni continue perché al termine del round non si annullano i voti precedentemente acquisiti da parte dei Witnesses, ma le votazioni vanno solo ad incrementare il punteggio dei Witnesses e quindi all'inizio del nuovo round i primi 20 Witnesses rimangono gli stessi del turno precedente a meno di cambiamenti nelle prime 20 posizioni della graduatoria. Dato il sistema di votazioni abbiamo che la maggior parte dei Witnesses rimarrà anche nei turni successivi, quindi per evitare che un Witness ignori un blocco inserito dal Witnesses in posizione precedente a lui per più round, ostacolando il suo lavoro, le posizioni dei Witnesses vengono mescolate a ogni round. Inoltre, abbiamo che se un Witnesses che è stato votato non ha inserito nessun blocco nelle ultime 24 ore sarà disabilitato fino a quando non aggiornerà la sua Block Signing Key.

La Block Signing Key è una coppia di chiavi (pubblica e segreta) che identifica un Witness da tutti gli altri utenti della rete. Infatti, quando un utente diventa Witness in automatico la sua coppia di chiavi Active key, che veniva utilizzata per le transazioni finanziarie, diventa la sua Block Signing Key che utilizza per firmare i blocchi da lui inseriti. A questo punto il Witness ha due scelte: utilizzare la Block Signing Key per entrambi gli scopi, ovvero sia per firmare le transazioni sia per firmare i blocchi, oppure se vuole mantenere un alto livello di privacy genera un'altra coppia di chiavi che utilizzerà unicamente come Active key.
Viene utilizzato questo protocollo perché fornisce alta scalabilità, infatti abbiamo l'inserimento di un blocco ogni 3 secondi, mantenendo anche un alto livello di sicurezza grazie anche alla struttura monetaria di Steem. Infatti, i Witnesses vengono pagati in SP e questo fattore non incentiva gli attacchi interni da parte dei Witnesses, anzi, come detto in precedenza, la moneta SP non può essere scambiata tramite Exchange e impiega 13 settimane per essere convertita in STEEM. Quindi se un Witness volesse attaccare la rete non avrebbe nessun compenso e questo aumenta il livello di sicurezza del protocollo.

Steem Witness Overview

#	Witness	Votes (MV)		Version	Price Feed	Produced	Missed	Missed %	Url	Vote
1	steemchiller	124,389.096		0.23.1	0.528 / 1.000	568,321	145	0.03 %	>>	
2	justyy	119,519.474		0.23.1	0.525 / 1.000	613,523	816	0.13 %	>>	
3	steem-agora	117,613.745		0.23.1	0.525 / 1.000	522,068	1,127	0.22 %	>>	
4	dev.supporters	115,049.652		0.23.1	0.525 / 1.000	589,652	3,313	0.56 %	>>	
5	future.witness	102,900.850		0.23.1	0.524 / 1.000	602,835	803	0.13 %	>>	
6	dlike	101,888.570		0.23.1	0.527 / 1.000	587,553	87	0.01 %	>>	
7	symbionts	97,587.894		0.23.1	0.526 / 1.000	555,181	270	0.05 %	>>	
8	rnt1	95,164.950		0.23.1	0.528 / 1.000	519,884	63	0.01 %	>>	

Figura 5.1: Nell'immagine si possono osservare alcune statistiche degli 8 witnesses che al momento dello screen hanno ricevuto più voti.

Capitolo 6

Compenso della rete

A partire dalla sedicesima hard fork del dicembre 2016, Steem crea nuovi token con un tasso di inflazione annuale del 9.5%. Il tasso diminuisce di circa 0.5% ogni anno. Il tasso continuerà a decrescere fino ad aver raggiunto lo 0.95%, dopo di che rimarrà costante per gli anni a seguire.
La quantità annuale di STEEM generata viene quindi divisa nel seguente modo:

- 15% ai possessori di SP;
- 10% ai Witnesses eletti che hanno inserito blocchi;
- 75% va Rewards Pool.

Il Rewards Pool è un fondo che viene utilizzato alla fine della giornata per ricompensare gli utenti che inviano, votano e discutono dei contenuti nella rete. La remunerazione di questi utenti avviene seguendo la legge di Zipf.
La legge afferma che dato una collezione molto grande di elementi, la popolarità del K esimo elemento, ordinato per popolarità, è 1/K della popolarità del primo elemento della stessa.
I creatori di Steem, affidandosi a questa legge empirica, stabilendo che la popolarità sia data dal numero di up-vote meno il numero di down vote, hanno stabilito che per ogni post o commento, il compenso di ogni elemento K, decresca in scala logaritmica.
Si utilizza questo metodo in modo che anche gli utenti meno popolari possano ricevere un qualche incentivo che, anche se piccolo, li porti a continuare ad investire il loro tempo nella comunità vedendo, giorno per giorno, che il loro contributo alla stessa sale con l'ammontare ricavato.

Capitolo 7

Transactions Fees

Uno degli aspetti più interessanti di Steem è l'aver elaborato una tecnica per annullare i costi di transazione. Secondo i creatori della piattaforma, una soluzione che richiede Fee per effettuare delle azioni che fino ad oggi sono state gratuite come la registrazione ad un social o poter postare un commento o esprimere il proprio giudizio su qualcosa, avrebbe impedito alla piattaforma di poter competere con le alternative centralizzate.
Nelle blockchain basate su Fee ad ogni transazione si applica un costo non nullo che anche se piccolo, impedisce che qualcuno possa richiedere un numero elevato di transazioni inutili a tal punto da saturare la capacità della rete e rendere inutilizzabile il servizio.
Steem propone il concetto di Fractional Reserve: ogni utente, in funzione della quantità di SP che possiede, ha diritto ad una banda garantita, cioè ad un numero minimo di transazioni garantite in un determinato arco temporale, in particolare si va a considerare la banda media settimanale.
È interessante notare come non si ponga un limite massimo di banda utilizzabile: se la rete fosse completamente scarica un utente avrebbe teoricamente la possibilità di saturare tutta la capacità della rete con le sue transazioni. Quando invece la rete è satura, se un utente stesse cercando di effettuare più transazioni di quanto la sua banda garantita gli permettesse, allora troverebbe le sue transazioni bloccate fino a quando o la rete torni in uno stato non saturo oppure fino a quando la sua banda media settimanale scenda al di sotto della soglia garantita.
Steem impone che ogni utente debba avere un saldo minimo di SP in modo da garantire la possibilità di effettuare un certo numero di transazioni anche in caso la rete fosse satura.
Interessante è ora capire come un utente che volesse registrarsi alla piattaforma possa ricevere la quantità minima di SP necessari.
Se la piattaforma regalasse SP ad ogni nuovo iscritto, ci si esporrebbe a Sybil Attack. Per questo è stato ingegnato un meccanismo per cui, al momento della registrazione, il quantitativo di Token necessari viene fornito da un utente già presente nella piattaforma, che li "presta" col diritto di riprenderseli in qualsiasi

momento e che frutterà al creditore un interesse sui guadagni dell'utente appena iscritto.
I token prestati saranno utilizzabili dall'utente iscritto come se fossero propri, potrà quindi utilizzarli per compiere azioni nel social, ma non potrà scambiarli per STEEM poiché non ne è il proprietario.
In pratica i Register (coloro che permettono la registrazione) a tutti gli effetti investono parte del loro SP in un nuovo utente, con la speranza che questo possa partecipare alla comunità e far fruttare il loro investimento.
Si intuisce come il Register abbia tutto l'interesse a concedere la somma solo ad utenti reali o, ancora meglio, ad utenti che potenzialmente partecipino attivamente alla piattaforma.

Capitolo 8

Conclusione

Steem si propone come un progetto che vuole cambiare radicalmente il modo in cui la blockchain viene percepita dall'utente finale.
Per definizione la piattaforma social che Steem punta ad essere necessita di scalabilità, moderazione dei contenuti, gratuità dell'utilizzo, immediatezza della risposta. Steem propone una soluzione basata sulla blockchain, tecnologia che appariva avere evidenti limiti in tutti questi aspetti e che, senza le geniali innovazioni proposte dai creatori, non avrebbe mai potuto portare a compimento una simile sfida.
Tramite la combinazione di idee innovative quali l'azzeramento dei costi delle transazioni e la garanzia di un ricavo da parte dei creatori e curatori per il lavoro svolto, e, grazie alle idee dietro SP e DPoS, Steem ha dimostrato la realizzabilità di una piattaforma social completamente distribuita. Inoltre, un simile traguardo dimostra che la Blockchain sia una tecnologia valida e pronta per il grande pubblico, che, anche se ancora con alcuni difetti, permette il superamento di limitazioni che le alternative del mondo centralizzato tendono ad avere.
La massima espressione di queste nuove potenzialità è il superamento della Censura.

Bibliografia

[1] Steem Bluepaper - https://steem.com/steem-bluepaper.pdf.

[2] Steem Whitepaper - https://steem.com/SteemWhitePaper.pdf.

[3] Steemit FAQ - https://steemit.com/faq.html#Where_do_the_new_STEEM_tokens_come_from.

[4] Steem Blockchain Tools - https://steemworld.org/

[5] Steemit's Permission Keys - https://steemit.com/steemit/@steemitguide/a-complete-guide-on-steemit-permission-keys-posting-owner-active-memo-digital-passwords-with-unique-functionality-that-allows

[6] Powering up and Powering Down - https://steemit.com/steempostit/@thenightflier/conversione-in-steem-powe

ETHEREUM ADDRESSES E STANDARD EIP-55

Enrico Guglielmino, Luca Giorgino, Carmen Frasca

Introduzione

Ethereum è stato fondato nel 2015 da Vitalik Buterin, programmatore russo diventato il più giovane cripto-miliardario del mondo all'età di 27 anni. Si tratta di una piattaforma globale, open source, pubblica e basata su una **Blockchain**. La caratteristica principale di Ethereum è quella di essere una piattaforma decentralizzata del Web 3.0 utile alla creazione e pubblicazione, su una rete di nodi peer-to-peer, di applicazioni decentralizzate (**dApps**), dette **Smart Contract** che sono create in un linguaggio di programmazione Turing-completo. Attualmente (dopo la nascita di Ethereum) con Smart Contract si intende un programma che viene messo in esecuzione sui nodi validatori di una blockchain il cui risultato rappresenta una transazione sulla quale i nodi validatori devono trovare un consenso. Gli smart contract aspirano ad assicurare una sicurezza e una velocità di esecuzione superiore alla contrattualistica esistente e di ridurre i costi di transazione associati alla contrattazione. Questi contratti sono spesso utilizzati per rappresentare altri beni come oggetti fisici del mondo reale (titoli di proprietà immobiliari) o oggetti puramente digitali (come i token). Ethereum si basa su una blockchain utilizzata per garantire che tutti i nodi eseguano la medesima applicazione e che i dati in input siano gli stessi, assicurandosi quindi indirettamente che anche l'output sarà lo stesso per tutti i nodi. Al fine di girare sulla rete peer-to-peer, i contratti di Ethereum "pagano" un utilizzo della potenza computazionale tramite una unità di conto, che viene detta Ether (**ETH**) , che funge quindi sia da criptovaluta che da carburante. Gli ETH possono essere utilizzati per le transazioni economiche e per fare trading, ma sono soprattutto uno strumento per gestire quello che è una sorta di computer globale, le cui applicazioni una volta lanciate non possono essere bloccate. Possiamo quindi dire che Ethereum gestisce i contratti in maniera intelligente nel senso che le applicazioni vengono eseguite esattamente in base a quanto programmato, senza alcuna possibilità di inattività, censura, frode o interferenze di terzi. Una delle tecnologie fondamentali di Ethereum è la crittografia, che può essere ad esempio utilizzata per dimostrare la conoscenza di un segreto senza tuttavia rivelarne il contenuto(attraverso una firma digitale) o per provare l'autenticità dei dati (attraverso un'impronta digitale). Questi tipi di prove crittografiche sono strumenti matematici fondamentali per il funzionamento della piattaforma Ethereum e sono inoltre ampiamente utilizzate nelle applicazioni Ethereum.

Capitolo 1

Ethereum addresses

Gli utenti di Ethereum possono essere anonimi, tuttavia i loro indirizzi lasciano una traccia pubblicamente visibile sulla blockchain. Gli indirizzi Ethereum sono identificatori univoci derivati da chiavi pubbliche o contratti ottenuti utilizzando la funzione hash unidirezionale **Keccak-256**. La curva utilizzata da Ethereum per generare gli indirizzi è la cosiddetta **secp256k1** (la stessa utilizzata da Bitcoin).

1.1 Chiavi private

Le chiavi private non sono utilizzate direttamente in Ethereum in alcun modo; esse non vengono mai trasmesse o archiviate sulla piattaforma. Una chiave privata è rappresentata da un numero scelto in maniera casuale. La proprietà e il controllo della chiave privata sono alla base del controllo di tutti i fondi associati all'indirizzo Ethereum corrispondente: perdere la chiave privata significa perdere per sempre i fondi relativi all'indirizzo Ethereum associato a quella chiave privata. La chiave privata viene utilizzata sia per creare la firma necessaria per spendere gli ether, sia per dimostrare la proprietà dei fondi in una transazione. La chiave privata deve quindi rimanere segreta, in quanto rivelarla a qualcuno significa dargli il controllo degli ether e dei contratti associati a quella chiave privata. Un semplice modo per generare una chiave privata in maniera random è il seguente:

- alle due facce di una moneta si associano rispettivamente i valori 0 e 1
- si lancia la moneta 256 volte
- si ricava la corrispettiva stringa binaria di 256 bit

Una volta ottenuta la chiave privata, quest'ultima può essere utilizzata in un Ethereum wallet. La chiave pubblica e l'indirizzo sono generati a partire dalla chiave privata.

1.2 Generazione di una chiave privata a partire da un numero random

Il primo e più importante passo nella generazione delle chavi è trovare una fonte sicura di entropia, o casualità. Creare una chiave privata di Ethereum implica la scelta di un numero intero compreso tra 1 e 2^{256}. Il metodo utilizzato per scegliere il numero non è rilevante, fintanto che non è prevedibile o deterministico. Ethereum utilizza un generatore di numeri casuali per produrre 256 bits casuali. Solitamente il generatore di numeri casuali è inizializzato con una fonte di casualità umana, motivo per il quale il sistema potrebbe chiedere di muovere il mouse per alcuni secondi o premere dei tasti casuali sulla tastiera. Un'alternativa valida potrebbe essere la rilevazione delle radiazioni cosmiche sul canale del microfono del computer.

Più precisamente, una chiave privata può essere qualsiasi numero diverso da zero fino a un numero molto grande leggermente inferiore a 2^{256}: un enorme numero di 78 cifre, circa $1.158 * 10^{77}$. Il numero esatto, che indichiamo con N, condivide le prime 38 cifre con 2^{256} ed è definito come l'ordine della curva ellittica utilizzata da Ethereum (secp256k1). Per creare una chiave privata, si sceglie casualmente un numero di 256 bits e si verifica (tramite la conversione binario-decimale) che sia compreso tra 1 e N. In termini di programmazione, il numero associato alla chiave privata si può ottenere a partire da una stringa ancora più grande di bits casuali (raccolti da una fonte di casualità crittograficamente sicura) di cui si calcola l'hash tramite l'algoritmo Keccak-256 che produce quindi un numero di 256 bits. Se il risultato rientra nell'intervallo valido, si ottiene una chiave privata adatta. Altrimenti, si riparte da una nuova stringa di bits casuale fino a quando non si ottiene un risultato valido.

2^{256}, la dimensione dello spazio delle chiavi private di Ethereum, è un numero incredibilmente grande. Di conseguenza, se si sceglie una chiave privata in maniera puramente casuale, è altamente improbabile che qualcuno possa mai indovinarla o scegliere la stessa chiave. Il processo di generazione della chiave privata è offline; esso non richiede alcun tipo di comunicazione con il network di Ethereum. Utilizzare un cattivo generatore di numeri causali (come la funzione rand pseudocasuale utilizzata nella maggior parte dei linguaggi di programmazione) è sicuramente una pessima scelta. Per tale motivo, per scegliere un numero che nessun altro sceglierà mai, è necessario che questo sia scelto in maniera veramente casuale. E' fondamentale utilizzare un generatore di numeri pseudocasuali crittograficamente sicuro (come CSPRNG) con un seme ottenuto da una fonte di sufficiente entropia. La corretta implementazione dello CSPRNG è fondamentale per la sicurezza delle chiavi.

L'accesso e il controllo dei fondi è realizzato tramite la firma digitale, la quale viene creata utilizzando la chiave privata. Le transazioni in Ethereum richiedono una firma digitale valida. Chiunque possieda una copia di una chiave privata ha il controllo del corrispondente account e anche degli ether che quest'ultimo detiene. Supponendo che un utente mantenga la propria chiave privata al sicuro, le firme digitali nelle transazioni di

Ethereum dimostrano il vero proprietario dei fondi, in quanto provano la proprietà della chiave privata.

1.3 Chiavi pubbliche

Una chiave pubblica di Ethereum è un punto della curva ellittica secp256k1, ed è quindi costituita da una coppia di coordinate (x, y) che soddisfano l'equazione della curva. La chiave pubblica è calcolata dalla chiave privata utilizzando la moltiplicazione su curve ellittiche, che è praticamente irreversibile:

$$K = k * G$$

dove k è la chiave privata, G è il generatore della curva e K è la chiave pubblica risultante. La moltiplicazione di k con G è equivalente a $G + G + ... + G$, (k volte). Dunque per produrre una chiave pubblica K da una chiave privata k, basta sommare il generatore G a se stesso k volte. Il generatore è specificato dallo standard secp256k1; è lo stesso per tutte le implementazioni di secp256k1, e tutte le chiavi derivanti da questa curva utilizzano lo stesso punto G. Poichè il generatore è sempre lo stesso per tutti gli utenti di Ethereum, una chiave privata k moltiplicata con G darà sempre come risultato la stessa chiave pubblica K. La relazione tra k e K è fissata, ma può essere solo calcolata in una direzione, da k a K. Questo è il motivo per cui un indirizzo Ethereum (derivato da K) può essere condiviso con chiunque senza tuttavia rivelare la chiave privata k dell'utente. Risalire dalla chiave pubblica alla chiave privata significa saper risolvere il problema del logaritmo discreto su curve ellittiche (ECDLP), problema attualmente non risolvibile computazionalmente. Nella maggior parte dei wallet le chiavi privata e pubblica vengono archiviate insieme per comodità. Tuttavia, la chiave pubblica può essere banalmente calcolata da quella privata, quindi è anche possibile memorizzare solo la chiave privata. In Ethereum le chiavi pubbliche sono solitamente rappresentate da una stringa di 130 caratteri esadecimali (65 bytes). Tale formato è stato proposto dal consorzio di industrie **Standards for Efficient Cryptography (SEC1)**. Lo standard definisce quattro possibili prefissi che possono essere utilizzati per identificare i punti sulla curva ellittica.

Prefix	Meaning	Length (bytes counting prefix)
`0x00`	Point at infinity	1
`0x04`	Uncompressed point	65
`0x02`	Compressed point with even y	33
`0x03`	Compressed point with odd y	33

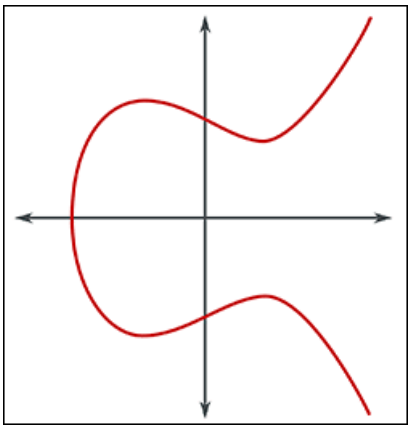
Ethereum utilizza solamente chiavi pubbliche uncompressed; di conseguenza l'unico prefisso esadecimale rilevante è 04. Lo standard prevede la concatenazione delle coordinate x e y della chiave pubblica:

$$04 + x\text{-coordinate (32 bytes/64 hex)} + y\text{-coordinate(32 bytes/64 hex)}$$

1.4 Secp256k1

La crittografia su curve ellittiche è un tipo di **crittografia asimmetrica** o a **chiave pubblica** basata sul **problema del logaritmo discreto**.

Figura 1.1:è un esempio di curva ellittica, simile a quella utilizzata da Ethereum.



Ethereum utilizza una curva ellittica specifica e un insieme di costanti matematiche, come definito nello standard secp256k1, stabilito dal **National Institute of Standards and Technology (NIST)**. La curva secp256k1 è definita dalla segunte funzione, che produce una curva ellittica:

$$y^2 = (x^3 + 7) \qquad \text{su } (F_p)$$

oppure

$$y^2 \bmod p = (x^3 + 7) \bmod p$$

Il mod p indica che questa curva è definita su un campo finito di ordine primo p, dove

$$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$$

Questa curva è definita su un campo finito di ordine primo invece che su i numeri reali, di conseguenza è costituita da un insieme (di cardinalità elevata) di punti sparsi in due dimensioni, il che rende difficile la visualizzazione. Tuttavia, la matematica è identica a quella di una curva ellittica sui numeri reali.

1.5 Keccak-256

Ethereum utilizza la funzione hash crittografica **Keccak-256** [1]. Keccak-256 è stata candidata per la **SHA-3 Cryptographic Hash Function Competition** indetta nel 2007 dal

[1]Ethereum utilizza Keccak-256, anche se nel codice viene spesso chiamato SHA-3

National Institute of Science and Technology. Keccak è stato l'algoritmo vincente, standardizzato come **Federal Information Processing Standard (FIPS-202)** nel 2015. Tuttavia, durante il periodo in cui Ethereum è stato sviluppato, la standardizzazione adottata dal NIST non era ancora stata finalizzata. Il NIST ha modificato alcuni parametri di Keccak dopo il completamento del processo di standardizzazione, presumibilmente per migliorare la sua efficienza. Alcune controversie derivanti dalla pubblicazione di documenti che implicavano degli interessi tra il NIST e la **National Security Agency** hanno portato a un significativo ritardo nella standardizzazione di SHA-3. All'epoca, Ethereum decise di implementare l'algoritmo Keccak originale, anzichè lo standard SHA-3 modificato dal NIST.

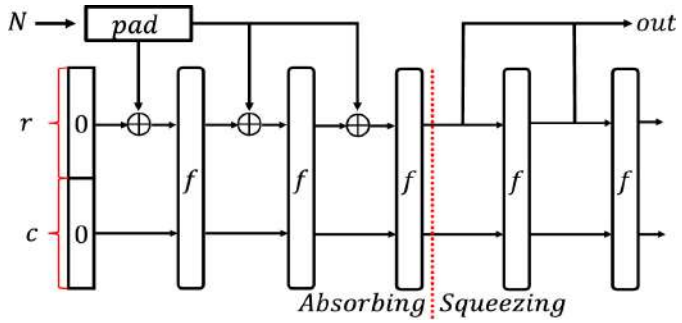
Come facciamo a sapere se la libreria che stiamo utilizzando implementa FIPS-202 SHA-3 o Kecck-256, dal momento che entrambi vengono chiamati "SHA-3"? Un semplice modo per capirlo è usare un vettore test, un output atteso per un dato input. Il test più comunemente usato riguarda l'utilizzo di una stringa vuota in input. Se si applicano le due funzioni hash alla stringa vuota si osservano i seguenti risultati:

Keccak256(" ") = $c5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85a470$
SHA3(" ") = $a7ffc6f8bf1ed76651c14756a061d662f580ff4de43b49fa82d80a4b80f8434a$

Indipendentemente da come viene chiamata la funzione, si può eseguire questo semplice test per capire se si tratta dell'originale Keccak-256 o dello standard NIST finale FIPS-202 SHA-3.

1.5.1 SHA-3

Vale la pena, dopo aver introdotto la funzione hash utilizzata da Ethereum, andare più nel dettaglio di come funziona. La differenza sostanziale con la precedente SHA-2 (precedenza che si deve intendere solo come temporale, in quanto SHA-2 è ancora utilizzata in vari contesti ed è considerata sicura) è che SHA-3 implementa una costruzione "a spugna", rappresentata da un blocco che prende in input un dato di qualsiasi dimensione (e quindi si ha la fase di assorbimento), e da in output una stringa di lunghezza fissa (la "spremitura" della funzione: si spreme fino a quando non si è raggiunto il numero giusto di bits in output). Dall'immagine si può avere un'intuizione di come funzioni, a grandi linee, la



costruzione a spugna. Il messaggio in input viene manipolato inizialmente tramite un padding per far sì che abbia una dimensione divisibile per r, dove r è il rate del sistema ed è il fattore che incide sulla velocità del sistema. Più è alto r, più veloce sarà l'algoritmo, quindi viene effettuato lo xor tra il singolo blocco che viene manipolato in quell'istante e lo stato corrente del sistema, che è diviso in due blocchi, uno di lunghezza r (ed è la porzione a cui viene applicato lo XOR), e uno di lunghezza c = b-r, che rappresenta la capacità del sistema. La funzione di permutazione f riceve un input di b bits e quindi opera sullo stato totale del sistema. Questo step di assorbimento viene effettuato fino a quando non vengono terminati i blocchi del messaggio. Quindi inizia la fase di spremitura del messaggio: la funzione f applica una permutazione all'input e restituisce un output di r bits: esegue lo stesso step fin quando non è stato raggiunto il numero di bits richiesti. Poiché la funzione f contiene solamente permutazioni, è facile calcolarne il livello di sicurezza, che è di $\frac{c}{2}$ (ovvero ci vogliono circa $2^{\frac{c}{2}}$ operazioni per romperlo). Inoltre la presenza dei bits di capacità all'interno dello stato rende SHA-3 resistente agli attacchi di tipo *length extension*: infatti l'output di Keccak-256 contiene solo una frazione dello stato interno, perciò un attaccante non può, avendo a disposizione la hash in output, concatenare a quella hash un altro messaggio contraffatto. Facendo uno zoom sulla funzione di permutazione, si può andare ad analizzare in dettaglio cosa succede all'interno di questa funzione e i vari passaggi che vengono eseguiti. In particolare, viene eseguito un certo numero di round (che dipende dalla dimensione del messaggio), e ogni round è composto da cinque step invertibili:

- θ per la diffusione: ad ogni bit dell'input (visto come una matrice 5*5*2^l) si somma la parità di due colonne: quella alla sinistra del bit in considerazione, e quella alla sua destra shiftata in avanti di una posizione
- ρ per la dispersione: la matrice viene tagliata lungo l'asse z, e ognuno di questi pezzi risultanti viene traslato ciclicamente di una certa quantità.
- π per scombussolare l'allineamento orizzontale/verticale di questi tagli: ogni taglio ruota con un certo periodo attorno ad un centro di gravità.
- χ è l'unico passaggio non lineare all'interno dei round: viene effettuato un calcolo bit per bit a livello di righe: preso un bit, viene mandato in xor con l'and logico tra i due bits che lo seguono, il primo dei quali negato.
- ι: si aggiungono delle costanti al messaggio per rompere la simmetria che era stata mantenuta nei passaggi precedenti.

Ognuno di questi passaggi viene ripetuto per $12 + 2 * l$ volte, dove 2^l è la dimensione in bit di ogni parola.

1.6 Indirizzi

Dopo aver presentato, a grandi linee, il funzionamento di Keccak-256, si può esaminare la sua applicazione principale all'interno di Ethereum, che riguarda la produzione di un indirizzo da una chiave pubblica. Un indirizzo Ethereum si costruisce nel seguente modo:

- ogni utente sceglie la propria chiave privata
- si deriva la corrispondente chiave pubblica K e si concatenano la coordinata x e la coordinata y scritte in esadecimale

  $K = 6e145ccef1033dea239875dd00dfb4fee6e3348b84985c92f103444683bae07b83b5c38e5e...$

- si utilizza Keccak-256 per calcolare l'hash della chiave pubblica

  $Keccak256(K) = 2a5bc342ed616b5ba5732269001d3f1ef827552ae1114027bd3ecf1f086ba0f9$

- si tengono solo gli ultimi 20 bytes (quelli meno significativi)

  $$001d3f1ef827552ae1114027bd3ecf1f086ba0f9$$

Nella maggior parte dei casi viene aggiunto il prefisso $0x$ che indica che l'indirizzo è codificato in esadecimale

$$0x001d3f1ef827552ae1114027bd3ecf1f086ba0f9$$

A differenze degli indirizzi Bitcoin, la cui codifica include un checksum integrato per proteggere gli indirizzi digitati in modo errato, gli indirizzi di Ethereum sono elaborati senza alcun checksum. La logica alla base di questa decisione era che un checksum, se necessario, può eventualmente essere nascosto e aggiunto ai livelli più alti del sistema. In realtà, questi strati superiori sono stati sviluppati troppo lentamente e questo ha portato a una serie di problemi nei primi giorni dell'ecosistema, compresa la perdita di fondi a causa di indirizzi errati e la validazione di input errati. Una delle soluzioni a questo problema è il protocollo ICAP (*Inter Echange Client Address Protocol*) che fornisce una codifica agli adress ethereum con un checksum abbastanza efficiente basato sullo stesso checksum che viene utilizzato nei codici IBAN delle carte di credito. L'**ICAP** è nato proprio per essere compatibile ed essere utilizzato come i famosi codici IBAN, riuscendoci solo in parte. Ovviamente il codice IBAN è un sistema centralizzato, mentre il corrispettivo di Ethereum è decentralizzato ma compatibile. Andando più sul tecnico, un codice IBAN è formato da una stringa di 34 caratteri alfanumerici, che identificano la nazione di appartenenza, il checksum e un identificatore dell'account bancario. Il codice ICAP, sulla sua falsa riga, introduce come codice "nazionale" la stringa "**XE**" che rappresenta "Ethereum", sguita da un checksum di due caratteri e poi tre possibili identificatori d'account:

- Diretto: un numero intero codificato in base 36 composto da 30 caratteri alfanumerici, che rappresentano i 155 bits meno significativi di un address ethereum. Il vantaggio di questa codifica è che è totalmente compatibile con i codici IBAN, sia in termini di lunghezza che di checksum, lo svantaggio è che non riesce a contenere tutti i 160 bits di un address Ethereum generico, perciò può essere implementato solo negli address che iniziano con uno o più bytes nulli.

- Basic: è simile alla codifica diretta, però è composto da 31 caratteri alfanumerici e quindi è possibile codificare qualsiasi tipo di address Ethereum, però è incompatibile con il codice IBAN perché è più lungo.
- Indiretto: Codifica un identificatore che "trova" il suo address Ethereum tramite un registro. É composto da 16 caratteri alfanumerici che comprendono un identificatore (ETH), un servizio che fornisce il registro e una stringa di 9 caratteri che comprende una parola di senso compiuto.

La validazione di un codice IBAN (e quindi anche di un codice ICAP, con le dovute variazioni sul tema) è molto semplice.

1. Controlla che la lunghezza del codice IBAN sia corretta.
2. Sposta i primi quattro caratteri (che corrispondono alla nazione e alle cifre del checksum) al fondo del codice.
3. Trasforma i caratteri alfabetici in caratteri numerici sfruttando questa trasformazione: A=10, B=11, ..., in modo da ottenere un numero intero.
4. Calcola il resto della divisione di questo numero per 97. Se il resto è 1, allora il codice è valido.

Riprendendo in mano l'indirizzo che è stato trovato prima

$$0x001d3f1ef827552ae1114027bd3ecf1f086ba0f9$$

si può osservare che inizia con degli zeri e quindi si può codificare in modo diretto.L'ICAP relativo a questo indirizzo è:

$$XE22HAMICDXSV5QXVJA7TJW47Q9CHWKJD$$

E adesso si può provare a validarlo secondo l'algoritmo specificato prima. Inanzitutto spostiamo le prime quattro cifre alla fine:

$$HAMICDXSV5QXVJA7TJW47Q9CHWKJDXE22$$

Quindi si trasforma questo codice in intero:

$$17102218121333283152633311910729193247269221732201913331422$$

e adesso si esegue la divisione per 97

$$17102218121333283152633311910729193247269221732201913331422 \equiv 1(mod 97)$$

Questa codifica ha però molti svantaggi, principalmente dovuti al fatto che esistono tre tipi di codifiche: diretto, indiretto e basic. Se un utente avesse un indirizzo che inizia con una stringa di zeri, e quindi fosse possibile codificarlo in modo diretto, se per caso digitasse un carattere alla fine dell'indirizzo, potrebbe risultare codificato come un indirizzo "basic" (che è formato da un carattere in più rispetto all'indirizzo diretto) e quindi potrebbe passare i controlli nonostante l'errore, perciò non è stato poi utilizzato e si è passati ad un altro standard.

1.7 EIP-55

EIP-55 standardizza gli indirizzi Ethereum assicurando che alcuni caratteri alfabetici dell'alfabeto diventino maiuscoli in modo tale da ridurre il rischio di inviare fondi all'indirizzo errato. Ma in che modo si effettua questa trasformazione da minuscolo a maiuscolo? Lo standard si basa sempre sulla solita funzione hash di Ethereum, Keccak-256. Si hasha l'indirizzo esadecimale minuscolo (che sarà il checksum), quindi si trasformano in maiuscolo tutti i caratteri alfabetici se la corrispondente cifra esadecimale dell'hash è maggiore di 8.

$$0x001d3f1ef827552ae1114027bd3ecf1f086ba0f9$$

Se convertito in standard EIP-55 diventa

$$0x001d3F1ef827552Ae1114027BD3ECF1f086bA0F9$$

Per capire come mai vengono modificati solo alcuni caratteri alfabetici calcoliamo la hash di questo indirizzo

$$23a69c1653e4ebbb619b0b2cb8a9bad49892a8b9695d9a19d8f673ca991deae1$$

Si può osservare che il carattere in sesta posizione della hash è una "c" che è maggiore di 8 in carattere esadecimale e quindi la corrispondente lettera dell'indirizzo (La "f") viene trasformata in lettera maiuscola. Con questo semplice trucco siamo al sicuro da errori di battitura. Infatti, se venisse digitato l'indirizzo sbagliato, il wallet, riconoscendo lo standard EIP-55 (perché trova delle lettere maiuscole), cerca di validarlo trasformando l'indirizzo tutto in lettere minuscole e calcolandone la hash. Se in corrispondenza di una lettera maiuscola è presente un numero inferiore ad otto, allora il wallet riconosce l'errore e impedisce di portare a termine l'operazione desiderata. Questo piccolo trucco è risultato sufficiente per risolvere il problema del checksum, infatti, secondo quanto racconta Vitalik, in media ci saranno 15 check bits per indirizzo, e la probabilità che un indirizzo scritto male possa superare il checksum di EIP-55 è dell'0.00247%, che è ben 50 volte migliore rispetto al checksum di ICAP, inoltre mantiene la lunghezza e la struttura dell'indirizzo identica. La maggior parte dei wallet che lavorano con Ethereum hanno ormai implementato questo standard, mentre quelli che non l'hanno ancora implementato semplicemente ignorano che ci siano alcuni caratteri maiuscoli.

1.8 Contract address

Come precedentemente anticipato oltre agli Externally Owned Account (EOA) esistono anche i Contract Account (CA) che rappresentano degli smart contracts. Ad ogni CA viene associato un indirizzo, che differentemente dagli EOA non è derivato da una chiave privata. Un contract address è calcolato deterministicamente dall'indirizzo del suo creatore e in base al numero di transazioni completate che ha inviato da quell'indirizzo al momento del deployment del contratto. Un esempio:

- dall'indirizzo *0x220a530fBBfE397C9F95279117fEf25e4490dA90*, con output transaction count (anche chiamato *nonce*) uguale a 2, viene fatto deployment di un contratto
- l'indirizzo e il nonce vengono concatenati e ne viene fatta la codifica *rlp* e in esadecimale il risultato è *d694220a530fbbfe397c9f95279117fef25e4490da9002*
- come per gli EOA, anche se con input diverso, viene calcolato l'hash

  $$Keccak256(rlp_{input}) = b6fb3c9e1165999e4de35978d196e1105e638d71ea0a03f902ccd3342e7bc0c2$$

- si tengono solo gli ultimi 20 bytes (quelli meno significativi) aggiungendo il prefisso *0x*, il risultato finale è il seguente

  $$ContractAddress = 0xd196e1105e638d71ea0a03f902ccd3342e7bc0c2$$

Note

- Il nonce è un valore scalare uguale al numero di transazioni inviate dall'indirizzo e/o al numero di contratti creati dall'account. Il nonce è un attributo dell'indirizzo mittente e ha senso solo nel suo contesto, non è memorizzato esplicitamente nella blockchain, ma è calcolato dinamicamente contando il numero di transazioni confermate che hanno origine da un indirizzo. Esempio sotto.
- Anche i contratti hanno i *nonces*, il nonce di un contratto viene incrementato solo quando da esso viene generato un altro contratto. Quando viene invocata una funzione del contratto, il nonce non viene incrementato, in quanto un CA è soltato il destinatario di una transazione.
- Dall'implementazione di EIP-161 il nonce di un contract address che è stato appena creato è inizializzato ad 1, nei contratti rilasciati prima di questo standard il nonce era inizializzato a 0.

1.9 Possibili implementazioni

1.9.1 Implementazione in Python

Di seguito viene presentata una possibile implementazione in Python di generazione della coppia delle chiavi (privata e pubblica), generazione degli indirizzi (externally owned accounts (EOAs) e contracts) e codifica EIP-55. Le librerie principali utilizzate sono:

- PyCryptodome per utilizzare la funzione di hash *Keccak256*
- ecdsa per la generazione di chiavi sulla curva *secp256k1*
- binascii per la conversione di dati in rappresentazione esadecimale a dato binario (o viceversa)
- rlp che implementa la Recursive Length Prefix encoding (RLP), codifica ampliamente utlizzata in Ethereum per serializzare gli oggetti.

```
1  from Crypto.Hash import keccak
2  from ecdsa import SigningKey,SECP256k1
3  from binascii import hexlify,unhexlify
4  import rlp, random
5
6  def private_key_gen():
7      return hexlify(SigningKey.generate(curve=SECP256k1).to_string())
8
9  def public_key_gen(k: str):
10     k = SigningKey.from_string(unhexlify(k),curve=SECP256k1)
11     K = k.get_verifying_key().to_string()
12     return hexlify(K)
13
14 def account_address_gen(K: str):
15     keccak_hash = keccak.new(digest_bits=256)
16     keccak_hash.update(unhexlify(K))
17     computed_hash = keccak_hash.hexdigest()
18     return '0x' + computed_hash[24:] #get the last 20bytes (40hex)
19 def contract_account_address_gen(address: str, tx_count: int):
20     address = address.replace('0x','')
21     # NOTICE that unhexlify return the same result with uppercase or
       lowercase
22     input_for_CA = rlp.encode([unhexlify(address),tx_count])
23     keccak_hash = keccak.new(digest_bits=256)
24     #get the last 20bytes (40hex)
25     contract_address = keccak_hash.update(input_for_CA).hexdigest()[24:]
26
27     # it is not EIPencoded, all hex digit are lowercase
28     return '0x' + contract_address
29
30 def EIP55_encode(address: str):
31     checksum = ""
32     address = address.replace('0x','').lower()
33
```

```
34      keccak_Hash = keccak.new(digest_bits=256)
35      #get the first 40 hex digit that correspond to 20 bytes
36      mask = keccak_Hash.update(address.encode()).hexdigest()[:40]
37
38      for i, digit in enumerate(address):
39          if digit in '0123456789':
40              # We can't upper-case the decimal digits
41              checksum += digit
42          elif digit in 'abcdef':
43              # Check if the corresponding hex digit in the hash is 8 or
    higher
44              if int(mask[i],16) > 7:
45                  checksum += digit.upper()
46              else:
47                  checksum += digit
48
49      return '0x' + checksum
50
51  def detect_EIP55_errors(address_to_check: str):
52      address_lower_case = address_to_check.lower()
53
54      checksum = EIP55_encode(address_lower_case)
55
56      if checksum != address_to_check:
57          return True # Errors found
58      else:
59          return False # No errors found
60
61  def stub_get_transaction_count(address: str):
62      # get number of not pending transaction for that address
63      return random.randrange(100)
64
65  def address_compare(addr_1: str, addr_2: str):
66      addr_1 = addr_1.replace('0x','').lower()
67      addr_2 = addr_2.replace('0x','').lower()
68
69      return addr_1 == addr_2
70  if __name__ == "__main__":
71      # example of usage
72      book_key = b'
    f8f8a2f43c8376ccb0871305060d7b27b0554d2cc72bccf41b2705608452f315'
73      k = private_key_gen()
74      K = public_key_gen(k)
75
76      address = account_address_gen(K)
77      eip55_address = EIP55_encode(address)
78      print(eip55_address)
79
80      if detect_EIP55_errors(eip55_address):
81          print("Error found")
82      else:
83          print("EIP55 compliant")
84
```

```
85      wrong_address = '0x001d3F1ef827552Ae1114027BD3ECF1f086bA0E9'
86      if detect_EIP55_errors(wrong_address):
87          print("Error found")
88      else:
89          print("EIP55 compliant")
90
91      random.seed()
92      nonce = stub_get_transaction_count(address)
93      contract_address = contract_account_address_gen(address,nonce)
94
95      forum_addr = '0x6ac7ea33f8831ea9dcc53393aaa88b25a785dbf0'
96      expected_contract_address = "0
        x343c43a37d37dff08ae8c4a11544c718abb4fcf8"
97      computed_contract_address = contract_account_address_gen(forum_addr
        ,1)
98
99      if address_compare(expected_contract_address,
        computed_contract_address):
100         print("Error in contract address generation")
101
102     metamask_account_address = '0
        x220a530fBBfE397C9F95279117fEf25e4490dA90'
103     private_key = b'
        dc9bcd6bc45712da0dc33b33292cd4a60e5deac1de1fc69fdc98ca3c68640450'
104     metamask_computed = EIP55_encode(account_address_gen(public_key_gen(
        private_key)))
105
106     print(metamask_computed)
107
108     faucet_contract = '0xd196e1105e638D71Ea0a03f902cCd3342E7bc0c2'
109     nonce = 2 # this is the tx_count
110     contract_computed = contract_account_address_gen(
        metamask_account_address,nonce)
111
112     if(address_compare(contract_computed,faucet_contract)):
113         print(nonce,contract_computed)
```

1.9.2 Note

Nel main sono presenti degli esempi di utilizzo delle funzioni implementate e sono anche stati utilizzati dei casi reali (esportando la chiave privata da un wallet MetaMask e facendo deploy di un contratto tramite Remix). Ciò ha permesso di verificare che le implementazioni delle funzioni sono corrette.

1.10 Interrogare la blockchain in Javascript

1.10.1 Come ottenere il numero di transazioni

Nelle poche righe seguenti viene prensentato come interrogare la blockchain per ottenere il numero di transazioni di diversi account (un EOA e due CA), vengono utilizzate le API di

Ethereum messe a disposizione dalla collezione di librerie web3.js che mette a disposizione diverse funzioni per interagione con un nodo ethereum, in questo caso ci si interfacciati con infura.io, è una piattaforma che mette a disposizione diversi strumenti di sviluppo per il testing e il deployment di applicazioni sulla blockcahin Ethereum.

```
let Web3 = require('web3');
const web3 = new Web3(new Web3.providers.HttpProvider('https://ropsten.
    infura.io/v3/33bc94be093043008a20f6b8fc65c576'))

let metamask_account = '0x220a530fBBfE397C9F95279117fEf25e4490dA90'
let contract = '0xd196e1105e638D71Ea0a03f902cCd3342E7bc0c2'
let contract_2 = '0xDA9Dd3bc865bd34aF3c5FAA2E6E16bf78a69CDa8'

web3.eth.getTransactionCount(contract)
.then(nonce=>console.log("CA1:",contract,nonce));

web3.eth.getTransactionCount(contract_2)
.then(nonce=>console.log("CA2:",contract_2,nonce));

web3.eth.getTransactionCount(metamask_account)
.then( nonce=>console.log("EOA:",metamask_account,nonce));
```

L'output ottenuto dall'esecuzione di questo script è il seguente:

```
CA2: 0xDA9Dd3bc865bd34aF3c5FAA2E6E16bf78a69CDa8 1
CA1: 0xd196e1105e638D71Ea0a03f902cCd3342E7bc0c2 1
EOA: 0x220a530fBBfE397C9F95279117fEf25e4490dA90 7
```

I risultati sono coerenti con quanto ci si aspettava in quanto l'indirizzo corrispondente all'EOA *0x220a530fBBfE397C9F95279117fEf25e4490dA90* ha effettuato 7 trasazioni di output, mentre i CA hanno nonce 1 dato che non hanno mai creato contratti (che è l'unico modo in cui possono incrementare il nonce come spiegato precedentemente). E' semplice verificarne la correttezza attraverso i seguenti link:

- https://ropsten.etherscan.io/address/0xd196e1105e638d71ea0a03f902ccd3342e7bc0c2
- https://ropsten.etherscan.io/address/0x220a530fbbfe397c9f95279117fef25e4490da90
- https://ropsten.etherscan.io/address/0xda9dd3bc865bd34af3c5faa2e6e16bf78a69cda8

Bibliografia

ethereum.stackexchange.com - how is the address of an ethereum contract computed? URL https://ethereum.stackexchange.com/questions/760/how-is-the-address-of-an-ethereum-contract-computed.

ethereum.stackexchange.com - do contracts also have a nonce?, a. URL https://ethereum.stackexchange.com/questions/764/do-contracts-also-have-a-nonce.

Yet another cool checksum address encoding, b. URL https://github.com/ethereum/eips/issues/55.

Inter echange client address protocol ICAP. URL https://eth.wiki/ideas/inter-exchange-client-address-protocol-icap.

Guido Bertoni. Crittografia simmetrica v 2.0, 10 Maggio 2019, Aula Buzano - Dipartimento di Scienze Matematiche, Politecnico di Torino. URL https://www.youtube.com/watch?v=gengmHxiXqY.

Andreas M. Antonopoulos e Gavin Wood. *Mastering Ethereum,Building Smart Contracts and Dapps*. O'Reilly, 2018.

COINJOIN
E MIXING DI BITCOIN

Alessandro Guggino, Carlo Iurato, Laura Marioni, Marco Momo

Indice

Capitolo 1

Introduzione

1.1 *Contenuti*

I problemi di privacy di Bitcoin sono noti, perciò questo scritto, a partire da una disamina di essi, si pone l'obiettivo di approfondire gli algoritmi ed i servizi di mixing come strumento per raggiungere un livello di privacy superiore.
La tecnica su cui si focalizza lo scritto è il CoinJoin, di cui si esaminano anche due sue implementazioni, sviluppate da JoinMarket e da Wasabi Wallet. Inoltre, si analizzano altre tecniche di mixing quali PayJoin, CoinSwap e TumbleBit.

1.2 *Privacy in Bitcoin*

Bitcoin è il primo sistema di denaro elettronico decentralizzato, ovvero che permette ai suoi utenti di scambiare valuta in modo sicuro senza dovere fare riferimento ad una autorità centrale. Il sistema è composto da una rete *peer-to-peer* di utenti e fonda la sua funzionalità sull'utilizzo di una Blockchain pubblica in cui le transazioni effettuate vengono registrate in maniera permanente, verificabile e distribuita.

I proprietari di criptovaluta utilizzano dei software o hardware, detti *wallet*, per gestire tutti i dati utili per effettuare transazione. In particolare, vengono memorizzarti gli *indirizzi bitcoin* ovvero i riferimenti sui quali i bitcoin possono essere depositati. Attraverso i wallet è possibile effettuare transazioni, ovvero spostare della valuta da un indirizzo ad un altro.
Quando un utente decide di effettuare una transazione deve scegliere:

- Gli *indirizzi di input* ovvero, da quali indirizzi, da lui posseduti, spendere l'ammontare richiesto;
- Gli *indirizzi di output* ovvero, verso quali indirizzi inviarlo;
- Come gestire il resto, qualora il valore totale contenuto negli indirizzi di input superasse quello richiesto dal ricevemente.

E' buona prassi fare sì che l'indirizzo di resto non sia tra quelli inseriti come input della transazione, perciò gli indirizzi Bitcoin sono in un certo senso usa e getta.

Le transazioni possono considerarsi validate e confermate dal sistema quando vengono registate sulla Blockchain. Quest'ultima è pubblica ed è visibile da ogni nodo della rete. Inoltre, esistono siti web in cui chiunque può esplorare la Blockchain in tutta la sua storia.
Il sistema perciò è del tutto trasparente, tutti sanno tra quali indirizzi la moneta è scambiata e in che quantità. Tuttavia Bitcoin si presenta come un sistema con un alto grado di privacy, ciò che permette questa qualità è il fatto che non vi è un collegamento esplicito tra indirizzi e l'identità reale dei proprietari.
Infatti, dal punto di vista di un osservartore qualsiasi, gli indirizzi Bitcoin non sono niente altro che stringhe di caratteri esadecimali casuali che non rivelano alcuna informazione su chi sia il vero proprietario. Inoltre, confrontare due indirizzi qualsiasi tra di loro non permette nemmeno di concludere se siano posseduti da una medesima persona.
A questo punto verrebbe da dire che se non sono gli utenti stessi a rivelare esplicitamente la propria entità, il sistema Bitcoin è del tutto privato.
In realtà non è proprio così, perchè si il sistema è ben struttuarato ma, sia i modi attraverso i quali la maggior parte degli utenti ne entrano a fare parte, sia i modi attraverso i quali gli utenti eseguono transazioni, presentano punti deboli a cui corrispondono vulnerabilità nella privacy.
Ecco due esempi:

1. Il modo comune per iniziare a utilizzare criptovaluta è comprarla dagli exchange, ai quali bisogna dare tutte le credenziali utili all'identificazione, proprio come si farebbe con una banca.

2. Scrivere ingenuamente una transazione può evidenziare la proprietà degli indirizzi. Considerando la transazione

   $$\{A : 1.05 \quad BTC\} \quad -> \quad \{B : 1 \quad BTC, \quad C : 0.05 \quad BTC\}$$

   è molto probabile che l'indirizzo C appartenga allo stesso proprietario dell'indirizzo A, poichè sembra proprio essere il resto dell'operazione.

Questi sono solamente due casi eclatanti, ma potrebbero essere i primi di un lunga serie comportamenti scorretti che portano a possibili violazioni della privacy, peraltro molti dei quali di largo uso comune.

In generale chi vuole violare la privacy Bitcoin lo può fare due modi

- **Attacchi on-Blockchain**,
- **Attacchi off-Blockchain**.

Con attacchi on-Blockchain si intendono quelle strategie che permettono di raggruppare insiemi di indirizzi appartenenti ad un medesimo proprietario, sfruttando le informazioni contenute nella Blockchain. Mentre gli attacchi off-Blockchain

sono quelli che permettono di risalire all'identità di un indirizzo sfruttando le informazioni derivanti dai software utilizzati per interfacciarsi con Bitcoin. L'utilizzo congiunto delle due strategie di attacco può portare a gravi violazioni della privacy degli utenti.
Nel seguito, si approfondirà che cosa si intende con questo tipo di attacchi, cercando di evidenziare quali sono i punti deboli per la privacy di Bitcoin. Dopodichè si elencheranno quali sono le strategie che si possono seguire per aumentare la privacy delle proprie operazioni e si proseguirà con l'approfondirne una categoria particolare: i *servizi di mixing*.

1.2.1 Attacchi on-Blockchain

Come si è introdotto in precedenza, in una transazione sono presenti:

- uno o più indirizzi di input,
- uno o più indirizzi di output.

E' chiaro che è possibile pagare più destinatari con la stessa transazione. Inoltre, più parti possono partecipare all'input di una transazione in quanto sono possibili transazioni multisig. Non è invece chiaro ad un osservatore esterno che relazione di proprietà sussiste tra gli indirizzi presenti in una transazione.
Gli attacchi on-Blockchain mirano ad estrarre proprio questa informazione. Per fare ciò, dato un insieme di transazioni, si crea il cosidetto **grafo delle transazioni**. Ovvero, un grafo i cui nodi sono gli indirizzi e in cui tra due indirizzi sussiste un arco se sono coinvolti in una stessa transazione. Questo oggetto permette di seguire la dinamica di insiemi di indirizzi nel corso del tempo. L'obiettivo di un attaccante è quello di estrarre i **wallet clusters**, ovvero insieme di indirizzi fortemente collegati tra di loro che pertanto possono essere ricondotti al medesimo proprietario. Una volta che i wallet clusters sono stati ottenuti, se l'attaccante in qualche modo riesce a venire a conoscenza di qualche informazione reale relativa a uno solo degli indirizzi del cluster, la privacy viene totalmente violata.
Vi sono varie tecniche per estrarre i wallet cluster da un grafo delle transazioni, di seguito ne vengono indicate solamente alcune. Queste consistono in semplici assunzioni e ragionamenti basati su aspetti comuni nelle transazioni.

Common-input-ownership euristic

Le transazioni più comuni sono quelle con *address single-signature*, pertanto è ragionevole assumere che tutti gli input di una transazione appartegano ad un medesimo wallet. CoinJoin vuole rompere questa euristica.

Change address detection

Consiste nell'analizzare la struttura della transazione per capire se e quando esiste un indirizzo di resto. Situazioni significative sono le seguenti:

- Quando verso la maggior parte degli output viene inviata una cifra tonda, i pochi indirizzi con cifre non tonde probabilmente sono da imputare ad indirizzi di resto;
- Se non vi è alcuna traccia di resto, probabilmente è segno che i Bitcoin non si sono mossi di proprietà, ovvero gli indirizzi di input e output sono della stessa persona.

Address reuse detection

Consiste nel verificare se tra gli indirizzi destinatari compaiono indirizzi già utilizzati in precedenza. Infatti, il riutilizzo di indirizzi già sfruttati in precedenza è una pratica banale che può mettere a rischio la privacy, per la semplice ragione che un attaccante può recuperare tutta la storia passata dell'indirizzo disponendo quindi di molte più informazioni per proseguire l'analisi. D'altra parte questa pratica è anche comune perchè gli utenti sono comodi a ad accumulare bitcoin su pochi indirizzi.

Cluster Growth

I cluster creati con queste tecniche crescono tendenzialmente in modo lento ed incrementale. L'unione di due grossi cluster è un segnale che l'euristica utilizzata è scorretta. Pertanto un altro modo di scovare l'indirizzo di resto è quello di sceglierne uno che permetta un incremento lento e graduale.

Tutte queste euristiche sono effettive fintanto che gli utenti scrivono le transazioni in maniera prevedibile e strutturata. Pertanto, il comportamento dell'utente può fare molto per impedire questo tipo di attacchi, nello specifico gli utenti devono cercare di scrivere le transazioni in modo tale che siano difficilmente interpretabili da un osservatore esterno.

1.2.2 Attacchi off-Blockchain

I nodi della rete Bitcoin comunicano attraverso una rete peer-to-peer per trasmettere transazioni e blocchi. Ciascun nodo trasmette le informazioni che gli giungono a tutti i nodi con cui comunica. Ciò è molto buono dal punto di vista della privacy, perchè i nodi destinatari non sanno stabilire se le transazioni che gli giungono sono state originate dai nodi vicini o se questi ultimi sono solamente dei ritrasmettitori dell'informazione.
Per entrare nella rete Bitcoin occorre utilizzare la rete internet e, a meno che non si usi Tor, un avversario (come ad esempio una società di sorveglianza delle transazioni) può spiare la connessione di un nodo e vedere le transazioni inviate e ricevute.
Attraverso attacchi di *analisi del traffico internet*, anche se la connessione che si utilizza è criptata, è possibile capire se un dato indirizzo IP è un nodo della

rete Bitcoin e, ancora più facilmente, capire se è un miner.
Stategie come il *Sybil attack* permettono addirittura di stabilire da dove è stata originata una transazione o un blocco. Questo attacco si basa sul creare tanti falsi nodi su diversi indirizzi IP che cercano di instaurarsi aggressivamente sulla rete, connetendosi a più nodi possibili. Questa alta connetività permette di localizzare le nuove transazioni trasmesse e di tracciarle al proporgarsi nella rete.

Oltre a queste tipologie di attacchi sofisticati ci possono essere violazioni della privacy dalle origini molto banali, di cui si propongono due esempi:

1. Alcuni utenti dopo avere effettuato delle transazioni vanno su siti di *Blockchain explorer* e cercano la transazione inviata con l'obiettivo di vedere se è stata confermata. Una volta trovata, continuano ad aggiornare la pagina periodicamente fino a che la transazione non ha raggiunto un buon numero di conferme. Questo comportamento è pessimo dal punto di vista della privacy, perchè questi siti web possono facilmente legare l'indirizzo IP dell'utente con la transazione ricercata. Pertanto, se si volesse verificare se una transazione è stata confermata, è consigliabile farlo attraverso un nodo della rete Bitcoin e non attraverso explorer online.

2. Due utenti si scambiano via e-mail in modo non criptato i propri indirizzi Bitcoin per iniziare a commerciare. La comunicazione viene intercettata.

3. Alcuni utenti possono rivelare volontariamente informazioni sensibili quando decidono di acquistare criptovaluta dagli *exchange*, perchè questi ultimi fanno controlli di identità e anti-frode, pertanto è necessario fornire informazioni come nome reale, residenza e occupazione. Tutti questi dati possono essere facilmente collegati agli indirizzi usati nelle future transazioni.

1.2.3 Soluzioni per migliorare la privacy

Come si è visto, nonostante l'efficienza del sistema Bitcoin, le vulnerabilità alla privacy sono molteplici. Il problema è piuttosto rilevante perchè non solo è negativo per gli utenti che vogliono poter commerciare in anonimità, ma lo è anche dal punto di vista del sistema, in quanto può ledere la *fungibilità*. Infatti, Bitcoin provenienti da indirizzi tracciati potrebbero essere ritenuti non altrettanto validi come quelli rimasti in completa anonimità.
Risulta perciò necessario sviluppare soluzioni robuste rispetto agli attacchi di de-anonimizzazione prima descritti, in modo da preservare la privacy del sistema.
In primo luogo il comportamento dell'utente ha una sua rilevanza, quindi essi dovrebbero essere informati su quali prassi dover seguire, e ciò sarebbe sufficiente per impedire gli attacchi più banali.

I **servizi di mixing**, che saranno approfonditi in questo elaborato, sono tecniche mirate all'offuscare i legami presenti tra indirizzi di input e output nelle

transazioni, permettondo al sistema di essere più resistente rispetto ad attacchi maggiormente sofisticati.
Oltre ai servizi di mixing, un'altra strategia rilevante è quella di utilizzare *Lighting Network*, un metodo di pagamento di secondo livello che permette di effettuare transazioni off-chain, ovvero senza doverle registare sulla Blockchain. Risulta perciò non violabile da attacchi on-blockchain, tuttavia è adatto solo per micropagamenti.

Capitolo 2

CoinJoin

CoinJoin è un protocollo implementato per migliorare la privacy e l'anonimato degli utenti di Bitcoin, aumentando così anche la fungibilità delle monete.
L'implementazione di questo metodo è particolarmente utile per evitare di tenere traccia dei fondi ricevuti o dei pagamenti effettuati nel sistema Bitcoin.
Lo sviluppatore Gregory Maxwell, nell'agosto del 2013, ha pubblicato un messaggio sul forum BitcoinTalk, dove ha spiegato lo sviluppo e il funzionamento della sua idea, evidenziando che non implica alcuna modifica del protocollo Bitcoin. La tecnica consiste nel combinare gli UTXO (Unspent Transaction Outputs) degli utenti che partecipano al CoinJoin in una grande transazione con più input e più output.

2.1 *Funzionamento di CoinJoin*

Il metodo più semplice per avviare una transazione CoinJoin è tramite un server dedicato, in modo centralizzato. Ogni utente che vuole svolgere un CoinJoin dovrà connettersi al server per specificare quali input e output la transazione dovrebbe includere, quindi il server creerà una grande transazione che unirà tutte le transazioni singole una volta firmate dagli utenti.
Sorge il problema che colui che controlla il server avrà accesso a tutti i dati forniti da singoli utenti. Per tale motivo sono disponibili anche soluzioni decentralizzate di CoinJoin, che consentono agli utenti di creare una transazione CoinJoin peer-to-peer, senza alcun intermediario centrale, oppure soluzioni centralizzate che utilizzano blind signature. Anche se potrebbe essere coinvolto un coordinatore, gli utenti non devono rinunciare alla custodia dei propri fondi.
Nelle transazioni di CoinJoin, le firme di ogni transazione singola rimangono indipendenti l'una dall'altra. Ogni firma necessaria è creata dal dispositivo del partecipante, quindi chiunque cerchi di collegare le firme non sarà in grado di cambiare la transazione o reindirizzare i BTC.
La transazione non è valida e non sarà accettata dalla rete finché tutte le firme non saranno fornite, e nessuno firmerà una transazione che non è di suo gradi-

mento. Una volta che tutti gli utenti firmano ogni singola transazione, verranno raggruppati nella stessa transazione CoinJoin. Questa verrà elaborata e convalidata dai miner, come una tradizionale transazione Bitcoin, per includerla all'interno della blockchain.

Attraverso il seguente esempio semplificato, si può vedere da vicino il funzionamento del CoinJoin.
Si supponga che Alice voglia trasferire 1 BTC dall'indirizzo A all'indirizzo C di Carol, mentre Bob vuole trasferire 1 BTC dall'indirizzo B all'indirizzo D di Dave. Per rendere privata questa transazione, entrambi decidono di utilizzare CoinJoin allo scopo di combinare i loro trasferimenti in un'unica transazione che ha due input (A e B) e due output (C e D).
CoinJoin prende le transazioni di Alice e Bob e le combina in una sola. Se le loro operazioni richiedono il resto, esso verrà considerato tra gli output anche un indirizzo dedicato. L'operazione di pagamento CoinJoin avverrà solo quando Alice e Bob firmano crittograficamente gli UTXO delle rispettive transazioni.

Per mitigare la possibilità che qualcuno capisca quali input e output siano collegati, il protocollo deve essere standardizzato in qualche modo. Poiché gli input non possono essere facilmente standardizzati, una soluzione potrebbe essere quella di rendere predefiniti gli output e impostare una denominazione minima. Per esempio, si potrebbe limitare l'output esattamente a 0,5 BTC, vedi Figura 2.1. Ciò renderebbe difficile capire quale input corrisponda a quale output e viceversa, poiché ogni output sarà al massimo del valore di 0,5 BTC. Sarà comunque possibile analizzare le transazioni di CoinJoin, grazie a degli strumenti open-source quali *CoinJoin Sudoku* e *Boltzmann*, come si vedrà in seguito.

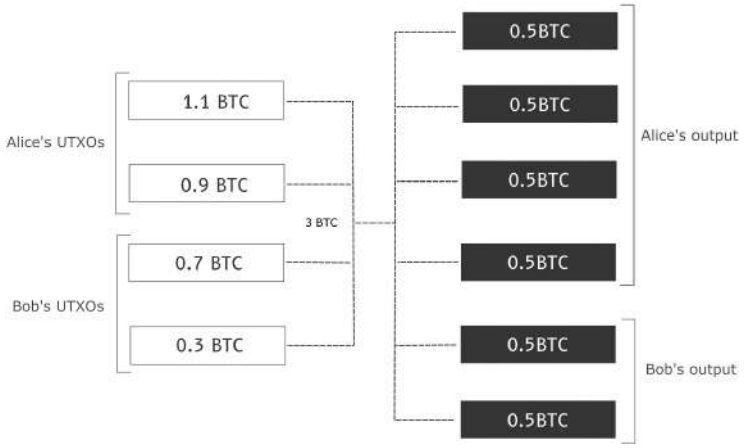


Figura 2.1: Esempio di CoinJoin

Anche il destinatario della transazione non è in grado di determinare da

quale indirizzo provengano i fondi ricevuti, perché i suoi UTXO non sono direttamente collegati ad un indirizzo singolo, ma ad una transazione con più input, potenzialmente indipendenti.

2.1.1 Sicurezza

CoinJoin non garantisce l'anonimato al 100%. Questo perché l'analisi dei dati, la trasparenza della blockchain e l'intercettazione delle informazioni in rete possono ancora essere utilizzati per violare la privacy in Bitcoin. Se però l'utente svolge una transazione CoinJoin con abbastanza entropia (ciò avviene quando molte persone partecipano alla transazione CoinJoin), la transazione si può considerare sicura.
L'insieme di utenti che potrebbero essere collegati a una transazione anonima costituiscono l'*anonymity set* per quella particolare transazione: quindi maggiore sarà la dimensione dell'anonymity set, maggiore sarà la sicurezza della transazione CoinJoin.
I sostenitori di questa tecnica sostengono che l'ideale sia un anonymity set di almeno 50 persone in un CoinJoin.

2.1.2 Costo di una transazione

Il costo di una transazione CoinJoin è molto più alto rispetto a quello di una comune transazione Bitcoin, poichè CoinJoin esegue *salti di mix*, cioé transazioni aggiuntive tra i partecipanti al CoinJoin. Ciascuno di questi salti genera una nuova transazione, che quindi ha una commissione associata. Più si desidera aumentare l'anonimato della propria transazione, maggiore sarà il numero di salti da fare, ottenendo così un costo finale maggiorato.

2.1.3 Vantaggi e svantaggi di CoinJoin

Vantaggi

- Migliora notevolmente la privacy delle transazioni;
- Permette di aumentare la fungibilità delle monete. Ad esempio, se i Bitcoin sono stati tracciati, il destinatorio potrebbe non accettare quei Bitcoin. CoinJoin permette di evitarlo, cioè fa sì che la moneta non perda il suo potere d'acquisto. Questa è una caratteristica molto importante considerando che l'ambiente Bitcoin è limitato a soli 21 milioni di monete;
- Non necessita di modifiche al protocollo per funzionare.

Svantaggi

- Può mescolare Bitcoin derivanti da attività illecite nascondendone la provenienza, perché non è specificato come vengono selezionati gli utenti che possono utilizzare il protocollo;

- Potrebbe esserci esposizione dei dati dell'utente se vengono uniti saldi usciti dal CoinJoin ad altri normali;

2.2 *Entropia di una transazione*

Gregory Maxwell suggerisce la nozione di "Entropia di CoinJoin", che misura quante combinazioni di input e output sono possibili.
L'entropia di una transazione è utilizzata per qualificare il grado di privacy fornito da una transazione e si definisce nel seguente modo:

$$E = log_2(N)$$

con:
- E = entropia della transazione
- N = numero di combinazioni (mappature di input e output).

Fondamentalmente, è l'entropia di Shannon con l'ipotesi che tutti gli eventi abbiano la stessa probabilità (cioè nessuna informazione aggiuntiva che dia una "preferenza" a specifiche combinazioni).
Si derivano diverse metriche che qualificano una transazione:

- **Entropia intrinseca**: è il valore calcolato senza alcuna informazione aggiuntiva, quando la transazione è considerata separatamente dalla blockchain;
- **Entropia effettiva**: è il valore che tiene conto delle informazioni aggiuntive;
- **Entropia massima**: è il valore associato ad una transazione CoinJoin perfetta che abbia una struttura simile o vicina alla transazione valutata.

L'entropia massima diventa molto più utile quando viene usata per calcolare l'efficienza del wallet e l'efficienza della blockchain.
L'**efficienza del wallet** si calcola nel seguente modo:

Efficienza del wallet = Entropia intrinseca - Entropia massima (in bit)

mentre l'**efficienza della blockchain**, che può essere utile per qualificare l'efficienza dell'intero ecosistema in termini di protezione della privacy degli utenti, si definisce come:

Efficienza della Blockchain = Entropia effettiva - Entropia massima (in bit)

L'entropia però non riesce a rilevare le perdite di privacy che si verificano a livelli inferiori, quindi vengono introdotte due metriche: la Link Probability (LP) di due UTXO e la Link Probability Matrix (LPM) di una transazione.

Link Probability e Link Probability Matrix

La Link Probability è la probabilità che un input abbia inviato dei fondi a un output e si definisce come:

$$LP(i,o,tx) = \frac{CombinationsWithLink(i,o)}{Combinations(tx)}$$

con:
- tx: una transazione bitcoin
- i: un input della tx
- o: un output della tx
- $Combinations(tx)$: Numero totale di combinazioni associate a tx
- $CombinationsWithLink(i,o)$: Numero di combinazioni di tx con un collegamento tra input e output.

La Link Probability Matrix di una transazione tx è una matrice che memorizza le probabilità di collegamento tra gli input e gli output della transazione e viene definita come:

$$LPM[m,n] = LP(I[n],\ O[m],\ tx)$$

con:
- I: insieme di input
- O: insieme di output

L'obiettivo è quello di avvicinare i valori all'interno della matrice a 0, cioè la probabilità che ci sia un legame tra l' m-esimo input e l' n-esimo output sia quasi nulla.

2.2.1 Implementazione di Boltzmann

Negli esempi sotto riportati si possono vedere gli output ottenuti analizzando transazioni test che hanno più input e più output.
Viene indicato il numero di combinazioni ottenibili da dati input e output e viene mostrato il valore dell'entropia della transazione: maggiore sarà il suo valore, maggiore sarà il livello di privacy della transazione.
Nel risultato si riporta, oltre all'efficienza del wallet, anche la Link Probability Matrix.
Questa implementazione in Python è un codice semplificato dove si utilizza un metodo di brute force per calcolare le differenti metriche ottenibili dati gli input e output.
Si potrà notare che all'aumentare dei partecipanti alla transazione, la probabilità di legame tra un input e un output tenderà a diminuire. Infatti, nel *TEST A* si hanno solo due input e non è difficile risalire alle possibili combinazioni tra input e output, mentre nel *TEST B* è più complicato risalire alle reali transazioni, infatti i valori delle probabilità nella Linkability Matrix sono diminuite e il numero di possibili combinazioni sono aumentate notevolmente.

```
—— TEST A ——————————————————————————————
Duration = 0.001971
Inputs = [('a', 10), ('b', 10)]
Outputs = [('A', 8), ('D', 8), ('B', 2), ('C', 2)]
Nb combinations = 5
Tx entropy = 2.321928
Wallet efficiency = 71.428571% (-0.485427 bits)
Linkability Matrix (probabilities) :
[[0.6 0.6]
 [0.6 0.6]
 [0.6 0.6]
 [0.6 0.6]]

—— TEST B ——————————————————————————————
Duration = 9969.582009
Inputs = [('a', 1), ('b', 1), ('c', 1), ('d', 1), ('e', 1),...
          ('f', 1), ('g', 1), ('h', 1), ('i', 1), ('j', 1), ('k', 1)]
Outputs = [('A', 1), ('B', 1), ('C', 1), ('D', 1), ('E', 1),...
           ('F', 1), ('G', 1), ('H', 1), ('I', 1), ('J', 1), ('K', 1)]
Nb combinations = 345322038293
Tx entropy = 38.329151
Wallet efficiency = 100.000000% (0.000000 bits)
Linkability Matrix (probabilities) :
[[0.18574595 0.18574595 0.18574595 0.18574595 0.18574595 0.18574595
  0.18574595 0.18574595 0.18574595 0.18574595 0.18574595]
 [0.18574595 0.18574595 0.18574595 0.18574595 0.18574595 0.18574595
  0.18574595 0.18574595 0.18574595 0.18574595 0.18574595]
 [0.18574595 0.18574595 0.18574595 0.18574595 0.18574595 0.18574595
  0.18574595 0.18574595 0.18574595 0.18574595 0.18574595]
 [0.18574595 0.18574595 0.18574595 0.18574595 0.18574595 0.18574595
  0.18574595 0.18574595 0.18574595 0.18574595 0.18574595]
 [0.18574595 0.18574595 0.18574595 0.18574595 0.18574595 0.18574595
  0.18574595 0.18574595 0.18574595 0.18574595 0.18574595]
 [0.18574595 0.18574595 0.18574595 0.18574595 0.18574595 0.18574595
  0.18574595 0.18574595 0.18574595 0.18574595 0.18574595]
 [0.18574595 0.18574595 0.18574595 0.18574595 0.18574595 0.18574595
  0.18574595 0.18574595 0.18574595 0.18574595 0.18574595]
 [0.18574595 0.18574595 0.18574595 0.18574595 0.18574595 0.18574595
  0.18574595 0.18574595 0.18574595 0.18574595 0.18574595]
 [0.18574595 0.18574595 0.18574595 0.18574595 0.18574595 0.18574595
  0.18574595 0.18574595 0.18574595 0.18574595 0.18574595]
 [0.18574595 0.18574595 0.18574595 0.18574595 0.18574595 0.18574595
  0.18574595 0.18574595 0.18574595 0.18574595 0.18574595]]
```

```
—— TEST C ——————————————————————————————
Duration = 0.097609
Inputs = [('b', 2), ('d', 1.5), ('a', 1), ('c', 0.5)]
Outputs = [('B', 2), ('C', 2), ('A', 1)]
Fees = 0
Nb combinations = 1
Tx entropy = 0.000000
Linkability Matrix (probabilities) :
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
('b', 2) & ('B', 2) are deterministically linked
('d', 1.5) & ('B', 2) are deterministically linked
('a', 1) & ('B', 2) are deterministically linked
('c', 0.5) & ('B', 2) are deterministically linked
('b', 2) & ('C', 2) are deterministically linked
('d', 1.5) & ('C', 2) are deterministically linked
('a', 1) & ('C', 2) are deterministically linked
('c', 0.5) & ('C', 2) are deterministically linked
('b', 2) & ('A', 1) are deterministically linked
('d', 1.5) & ('A', 1) are deterministically linked
('a', 1) & ('A', 1) are deterministically linked
('c', 0.5) & ('A', 1) are deterministically linked

—— TEST D ——————————————————————————————
Duration = 0.022586
Inputs = [('b', 2), ('d', 1.5), ('a', 1), ('c', 0.5)]
Outputs = [('A', 1), ('D', 1), ('E', 1), ('F', 1), ('B', 0.5), ('C', 0.5)]
Nb combinations = 759
Tx entropy = 9.567956
Wallet efficiency = 14.152527% (-2.820868 bits)
Linkability Matrix (probabilities) :
[[0.54808959 0.33333333 0.33333333 0.2173913 ]
 [0.54808959 0.33333333 0.33333333 0.2173913 ]
 [0.54808959 0.33333333 0.33333333 0.2173913 ]
 [0.54808959 0.33333333 0.33333333 0.2173913 ]
 [0.33860343 0.33860343 0.33860343 0.33860343]
 [0.33860343 0.33860343 0.33860343 0.33860343]]
```

Capitolo 3

Implementazioni di CoinJoin

3.1 *JoinMarket*

JoinMarket è un'implementazione di CoinJoin decentralizzata ed open-source che crea un nuovo tipo di mercato, permettendo agli utenti di effettuare una transazione CoinJoin e di scegliere il loro ruolo nella creazione di essa.
Da un lato ci sono i *market maker*, utenti che si occupano di organizzare il CoinJoin, creando un'offerta, e per questo possono decidere se ricevere delle fee dagli altri partecipanti. Dall'altro lato ci sono i *market taker*, utenti che possono partecipare subito al CoinJoin scegliendo tra le offerte disponibili, pagando una fee ove richiesta.

Siccome i market makers possono ricevere delle fee, gli utenti che possiedono grandi quantità di Bitcoin (e non sono intenzionati a venderli) possono usare JoinMarket per generare profitto da essi. È un investimento a basso rischio in cui si mette una quantità di Bitcoin in un hot wallet a disposizione del mercato e si agisce da market maker creando le offerte di CoinJoin. Questa operazione può essere svolta automaticamente tramite uno script chiamato *Yield Generator*. In tal modo si ha un guadagno economico ma anche di privacy e fungibilità dei Bitcoin posseduti. Il grande svantaggio a livello di sicurezza di Joinmarket è dato dal fatto che il wallet è online.

La privacy viene notevolmente migliorata ripetendo i CoinJoin più volte, perciò JoinMarket ha sviluppato uno script chiamato *Tumbler*, dove i CoinJoin sono creati automaticamente in momenti casuali per importi casuali ed i Bitcoin depositati nel wallet sono mixati tramite molti CoinJoin ed inviati a tre o più indirizzi di destinazione, senza effettuare il riutilizzo degli indirizzi. La caratteristica di utilizzare più di un indirizzo di destinazione è necessaria per

evitare la correlazione dell'importo.

Per evitare che una transazione successiva ricombini un change output con un CoinJoin output, siccome ciò fornirebbe la prova che i due UTXO siano di proprietà della stessa persona, il wallet di JoinMarket utilizza il concetto di *mixdepth*. Ogni mixdepth è un'identità diversa, per cui gli UTXO di un certo mixdepth non sono mai usati come input insieme a quelli di un altro mixdepth, ottenendo così un isolamento delle monete. Le monete possono muoversi tra mixdepth attraverso i CoinJoin, mentre il change output rimane nella stessa mixdepth.

Grazie ai recenti aggiornamenti nel 2020, JoinMarket supporta anche le transazioni di PayJoin.
Inoltre, è stato introdotto il *fidelity bond*, un meccanismo che permette ai market maker di congelare una quantità di Bitcoin al fine di avere maggiori probabilità di essere scelti dai market taker. Con questo incentivo, JoinMarket migliora la propria resistenza ai sybil attacks.

3.2 *Wasabi*

Wasabi è un wallet focalizzato sulla privacy che fornisce ai propri utenti un'implementazione centralizzata ed open-source di CoinJoin.
Wasabi permette un CoinJoin trustless (nessuno può rubare) e privato (nessuno, nemmeno il coordinatore - il server di Wasabi che gestisce il CoinJoin - può spiare) tramite la Schnorr blind signature. Questo avviene seguendo *ZeroLink*, un protocollo i cui autori sono due sviluppatori, uno di Wasabi Wallet e l'altro di Samourai Wallet. Il protocollo è stato poi forkato e modificato da Samourai per implementare il proprio servizio chiamato *Whirlpool*, che non sarà approfondito.

Wasabi, oltre al CoinJoin, permette anche il PayJoin, ed è impegnato da inizio 2020 nello sviluppo di un nuovo protocollo di CoinJoin chiamato *WabiSabi*.

3.2.1 ZeroLink

Il protocollo ZeroLink è composto da varie fasi:

Registrazione degli input

L'utente seleziona quali monete vuole registrare per il CoinJoin e Wasabi genera una *input proof*, ovvero una firma su un messaggio di challenge con la chiave privata che blocca le monete selezionate. In questo modo il coordinatore può verificare che l'utente possieda effettivamente le monete. Dopodiché, il client di Wasabi genera diversi nuovi indirizzi di output, a seconda del valore degli input registrati, i quali non devono essere collegabili agli input e perciò sono nascosti

tramite blind signature. Siccome il change output può essere collegabile facilmente agli input, il suo indirizzo viene mantenuto in chiaro.

Per ogni round, il client genera per l'utente una nuova identità su Tor - lo strumento che permette una comunicazione anonima su Internet - che non sia collegata alla connessione precedente, tramite cui invia delle informazioni al coordinatore:

- Le monete in input che si vogliono registrare, insieme alla input proof;
- Il change address in chiaro;
- L'output address cifrato.

Ora il coordinatore verifica che:

- Ci siano posti disponibili nel CoinJoin;
- L'output address (cifrato) non sia mai stato registrato prima;
- Ogni input non sia mai stato registrato prima, non sia bannato, non sia stato speso e che l'input proof sia valida;
- La somma degli input sia più alta del minimo valore richiesto.

Dopo aver terminato con successo i controlli, il coordinatore firma il blinded output e lo invia all'utente insieme ad un identificatore dell'identità su Tor utilizzata durante il round. L'utente è in possesso dei segreti per fare l'unblind del signed blinded output ritornato, rivelando così il suo output address e mantenendo la firma del coordinatore.

Questa fase finisce quando il numero degli input registrati supera il numero degli input richiesti o quando il tempo passato dal round precedente è di un'ora.

Conferma della connessione

Si verifica che tutti gli utenti siano online e pronti a continuare, così il coordinatore richiede gli identificatori delle identità agli utenti e, mentre è ancora in comunicazione, ritorna l'hash di tutti gli input registrati nel round.
Il round viene abbandonato se troppi utenti si sono disconnessi.

Questa fase termina quando tutti gli utenti hanno fornito il loro identificatore o dopo un intervallo di tempo se il numero degli utenti online è comunque maggiore di quelli richiesti.

Registrazione degli output

Il client genera per l'utente una nuova identità su Tor, tramite cui si inviano al coordinatore:

- L'output address in chiaro;
- La firma del coordinatore sull'output address;
- L'hash di tutti gli input registrati nel round.

Siccome il coordinatore può verificare la sua firma, è a conoscenza che l'output address è stato inviato da qualcuno durante la prima fase, e perciò è stato verificato, ma non può collegare le due identità.

Questa fase termina quando il valore negli output address più quello nei change address è uguale al valore degli input. Se dopo un intervallo di tempo alcuni utenti non hanno registrato il loro output address, questi sono bannati e viene iniziato un nuovo round.

Firma

Il coordinatore costruisce la transazione di CoinJoin con gli input address, output address, change address ed un output per la fee del coordinatore. La transazione viene inviata a tutte le identità della prima fase che devono verificare:

- L'uguaglianza tra l'hash degli input registrati e l'hash degli input nella transazione;
- La correttezza dei propri indirizzi di input e di output.

Dopodichè l'utente firma la transazione con le chiavi private dei suoi input ed invia l'identificatore, la firma e l'indice dell'input al coordinatore, che verifica le informazioni.

Questa fase finisce quando il coordinatore ha ricevuto tutte le firme valide per tutti gli input registrati.

Trasmissione

La transazione di CoinJoin è stata costruita e firmata, perciò è pronta: il coordinatore la invia tramite la rete Tor a dei nodi Bitcoin casuali.
Wasabi imposta delle mining fee basse, quindi la transazione sarà confermata in circa 12 ore, ma si possono comunque registrare gli output non confermati per un altro round di CoinJoin.

3.2.2 WabiSabi

WabiSabi è il nuovo protocollo che Wasabi sta sviluppando per il rilascio della seconda versione del proprio wallet, che probabilmente accadrà a fine 2021. Esso permetterà di fare CoinJoin più velocemente, con più privacy, con UTXO ridotti e privi di change output, ed inoltre avrà maggiore efficienza in termini di blockspace, quindi le transazioni di CoinJoin peseranno di meno e richiederanno meno mining fee.

Seppure WabiSabi non rimuoverà il coordinatore, vedendolo come necessario per rendere il protocollo il più possibile a bassa latenza, cercherà di ridurne i privacy leaks. Il nuovo protocollo permetterà agli utenti di prendere parte ad un CoinJoin con qualsiasi quantità di Bitcoin e non con una definita ed uguale per tutti i partecipanti. Un ulteriore miglioramento è legato al fatto che questa implementazione non sarà limitata all'auto-invio di monete ma consentirà anche pagamenti verso un altro utente.

Tra le novità crittografiche introdotte da WabiSabi vi sono i *Pedersen commitment*, utilizzati anche da Monero, e le *Keyed-Verification Anonymous Credential (KVAC)*, sviluppate ed utilizzate dall'applicazione di messaggistica criptata Signal per le chat di gruppo, al fine di rendere le informazioni dei membri disponibili ai soli membri del gruppo e nascoste a chi ne è esterno, tra cui il service provider (nel caso di Wasabi: il coordinatore).

Capitolo 4

Altri Mixing

Il coin mixing può fare riferimento a qualsiasi attività che coinvolge l'offuscamento di fondi sostituendoli con altri. Nel contesto delle criptovalute, il coin mixing indica comunemente un servizio fornito da terze parti.
Di solito, i fornitori del servizio prendono le monete degli utenti (e una piccola commissione), e restituiscono monete che non hanno collegamenti con quelle inviate. Questi servizi sono anche conosciuti come **tumbler** o **mixer**.
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. In fondo, gli utenti rinunciano al controllo dei loro Bitcoin nella speranza di ricevere indietro monete non collegate.

Nelle successive sezioni si prendono in esame tre servizi distinti: PayJoin, CoinSwap, TumbleBit.
Il primo è un approccio utilizzato per transazioni tra un commerciante e un cliente, che tuttavia presenta problemi nel caso in cui l'avversario del cliente, in termini di privacy, sia proprio il commerciante.
La seconda tecnica di mixing è nuova e non è ancora stata distribuita, perché molto complessa da implementare, tuttavia c'è grande interesse intorno ad essa perché offre grandi potenzialità.
Il terzo è un servizio di mixing centralizzato funzionante e compatibile con il protocollo Bitcoin. Esso si basa sulla risoluzione di puzzle crittografici per la riuscita delle transazioni.

4.1 *PayJoin*

PayJoin è una particolare tipologia di CoinJoin utilizzata da due entità, che prevede uno scambio di denaro da parte della prima verso la seconda. Questa

tecnica porta un miglioramento della privacy per Bitcoin. La transazione non ha gli output multipli distinitivi con gli stessi valori, e perciò non è ovviamente visibile come una transazione equal-output CoinJoin (Figura 4.1). Come visibile nell'esempio in figura, in questa tipologia di transazione è possibile affermare con buona certezza che l'output D è il resto appartenente al possessore dell'input Y, mentre l'output C è il resto appartenente al possessore dell'input X.

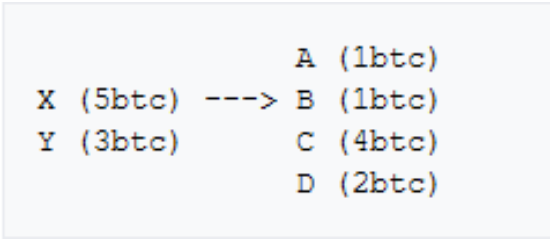


Figura 4.1: Esempio di transazione equal-output CoinJoin

Al contrario una transazione PayJoin si confonde perfettamente con una normale transazione Bitcoin con resto. L'esempio in Figura 4.2 mostra una transazione che può essere interpretata come un generico pagamento Bitcoin con il relativo resto. Un'altra possibile interpretazione potrebbe essere quella in cui l'input da 2 BTC è posseduto dal commerciante e quello da 5 BTC è posseduto dal cliente, e che la transazione sia il pagamento di 1 BTC dal compratore al venditore. Non c'è modo di sapere quale delle due interpretazioni sia quella corretta. Il risultato è quindi una transazione CoinJoin che rompe l'euristica common-input-ownership e migliora la privacy, ma è anche non rilevabile e indistinguibile da una normale transazione Bitcoin. Inoltre, non è possibile risalire a chi spettino i resti, come nel primo esempio visto in questo paragrafo.
Se si iniziasse a fare maggiore uso delle transazioni PayJoin, allora l'euristica common-input-ownership diverrebbe completamente inutilizzabile dalle compagnie che effettuano la sorveglianza delle transazioni, perchè non sarebbe più possibile affermare con certezza che due input di una transazione appartengono allo stesso soggetto e, inoltre, sarebbe molto più complesso stabilire a chi appartiene l'output relativo al resto, nel caso ci fosse. Proprio per questo motivo c'è grande interesse intorno all'idea di PayJoin.

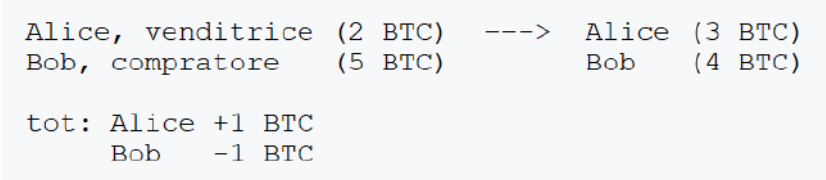


Figura 4.2: Esempio di transazione PayJoin

4.2 *CoinSwap*

CoinSwap è una tecnica per aumentare la privacy delle transazioni Bitcoin basata sul concetto di Atomic Swap, ideata come CoinJoin nel 2013 da Greg Maxwell. Con l'espressione *Alice e Bob effettuano un CoinSwap* si intende: Alice vuole scambiare un quantitativo di Bitcoin in suo possesso per lo stesso importo (meno le commissioni) dei Bitcoin di Bob, il tutto utilizzando degli smart-contract, in modo tale che nessuna delle due parti possa imbrogliare l'altra. In tal senso è molto importante il concetto di atomicità, poiché si fa in modo che uno scambio di criptovaluta sulla blockchain avvenga o meno, senza possibilità di frode.
CoinSwap rompe il grafo della transazione tra i Bitcoin inviati e ricevuti, facendo apparire lo scambio sulla blockchain come due insiemi di transazioni completamente separate.

```
Alice's Address ---> 2of2 multisig escrow address 1 ---> Bob's Address
Bob's Address   ---> 2of2 multisig escrow address 2 ---> Alice's Address
```

Figura 4.3: Esempio di CoinSwap

Come si può vedere nell'esempio in Figura 4.3, Alice e Bob possono scambiarsi le loro monete mandandole prima ad un indirizzo di CoinSwap, che poi le invierà ad un indirizzo di Bob. La privacy indubbiamente migliora, basta considerare che un osservatore della blockchain non può ricollegare i due indirizzi di Alice, poiché non c'è nessuna transazione diretta tra essi. CoinSwap in questo senso rompe l'euristica del grafo delle transazioni, per la quale si assume che: data una transazione da A a B, allora la proprietà dei fondi è andata da A a B. Tuttavia CoinSwap da sola non può migliorare di molto la privacy, perciò richiede altri strumenti per creare un sistema realmente privato.
La prima idea di CoinSwap richiedeva l'uso di una firma multipla 2 di 2 (2-of-2 multisig). Si è notato però che usando una firma multipla 2 di 3 (2-of-3 multisig) è possibile aumentare leggermente l'anonimato, grazie all'utilizzo di una terza chiave pubblica fittizia.

Dato che le transazioni si miscelano con le restanti di Bitcoin, un'applicazione basata su CoinSwap garantirebbe un grado maggiore di privacy rispetto alle esistenti applicazioni equal-output CoinJoin (JoinMarket, Wasabi Wallet). Inoltre CoinSwap sarebbe più economico a parità di privacy, poiché gli utenti di CoinJoin solitamente creano più CoinJoin per avere una privacy adeguata.

4.2.1 CoinSwap: Liquidity Market

È possibile creare un liquidity market per CoinSwap che sia molto simile a come funziona JoinMarket per CoinJoin. Nell'esempio in Figura 4.3, Alice sarebbe il *market taker*, mentre Bob sarebbe il *market maker*, quindi Alice paga una commisione a Bob per aver scelto il CoinSwap. In questo modo si ha una

migliore esperienza utente: Alice può in creare CoinSwap in qualunque momento per ogni somma desiderata.

4.2.2 CoinSwap multi-transazione

Una transazione CoinSwap come quella in Figura 4.3 è vulnerabile ad attacchi di tipo *amount correlation*: l'avversario inizia a tracciare le transazioni a partire dal primo indirizzo di Alice e, cercando all'interno della blockchain transazioni con lo stesso importo, arriva all'indirizzo finale di Alice. Per mitigare questa vulnerabilità è necessario creare una multi-transazione CoinSwap (Figura 4.4).

```
AliceA (15 BTC) ----> CoinSwap AddressA ----> BobA (15 BTC)

BobB (7 BTC) ----> CoinSwap AddressB ----> AliceB (7 BTC)
BobC (5 BTC) ----> CoinSwap AddressC ----> AliceC (5 BTC)
BobD (3 BTC) ----> CoinSwap AddressD ----> AliceD (3 BTC)
```

Figura 4.4: Esempio di multi-transaction CoinSwap

In questo caso Alice crea una transazione di 15 BTC e riceve tre transazioni che, sommate, raggiungono lo stesso importo. Ma siccome nella blockchain non c'è nessuna transazione ricevuta da Alice con importo di 15 BTC, non sarà possibile collegare l'indirizzo inziale di Alice a quello finale.

4.2.3 Routed CoinSwap

In una transazione CoinSwap tra due utenti come quella in Figura 4.3, Alice deve fidarsi di Bob, che sa esattamente dove sono le monete di Alice e potrebbe spiarla. Bob quindi rappresenta il *single point of failure* (unico punto di rottura) del protocollo. Per decentralizzare la fiducia si possono creare più CoinSwap in cui il pagamento di Alice è instradato tramite più utenti tipo Bob come in Figura 4.5.

```
AliceA ====> Bob ====> Charlie ====> Dennis ====> AliceB
```

Figura 4.5: Esempio di Routed CoinSwap

Il simbolo ====> rappresenta una transazione CoinSwap. In questo caso Alice sarà il *market taker* nel liquidity market e tutte le altre entità saranno i *market makers*. Solo Alice conoscerà l'intera route, mentre i market makers conosceranno solo gli indirizzi Bitcoin precedente e successivo nella route.

Combinando Routed CoinSwap con le multi-transazioni è possibile avere i benefici di entrambe le tecniche in termini di privacy. Si rende in questo modo difficilmente tracciabile il movimento di monete all'interno della rete.

4.2.4 PayJoin con CoinSwap

CoinSwap può essere utilizzato insieme a CoinJoin. Nella versione originale di CoinSwap, Alice può pagare ad un indirizzo CoinSwap utilizzando più indirizzi in suo possesso come input. In questo modo si potrebbe dedurre che tutti gli input appartengono alla stessa entità, avendo così perdita di privacy. Se si coinvolgesse Bob nella transazione, facendo inserire anche a lui un input, si romperebbe l'euristica common-input-ownership. Questo protocollo non ha caratteristiche particolari o pattern che lo possano rendere riconosibile o distinguibile da una normale transazione Bitcoin. Per questo motivo si avrebbe un aumento considerevole in termini di privacy.
Ci sono tuttavia delle differenze sostanziali tra PayJoin e CoinSwap. PayJoin funziona quando le due entità sono un commerciante e un cliente che cooperano per alzare la privacy di entrambi. L'avversario di entrambi, in questo caso, è un osservatore passivo della rete. PayJoin non aiuta il cliente se l'avversario è il mercante stesso, situazione che può capitare spesso. CoinSwap può aiutare nella risoluzione di questa situazione, perché non richiede che l'altra parte sia un amico (o un commerciante fidato nel caso dell'esempio).

4.2.5 CoinSwap: conclusioni

Essendo CoinSwap una tecnica relativamente complicata da implementare, fino al 2020 non era stata ancora distribuita. Il primo test di CoinSwap è stato eseguito recentemente, a dicembre 2020, da Chris Belcher. Tuttavia, ci sono ancora migliorie da apportare al protocollo affinché una transazione CoinSwap sia indistinguibile da una normale transazione Bitcoin.

4.3 *TumbleBit*

TumbleBit è un protocollo di anonimato per pagamenti Bitcoin, nasce nel 2016 ed è completamente compatibile con il protocollo Bitcoin attualmente in uso. Esso consente alle parti di effettuare un pagamento attraverso un Tumbler non fidato. Nessuno, nemmeno il Tumbler, può dire chi ha pagato e chi ha ricevuto il pagamento. TumbleBit è costituito da due protocolli di scambio equo interconnessi, che impediscono il furto di Bitcoin da parte di utenti o da parte del Tumbler. Questo è realizzato combinando calcoli crittografici veloci (eseguiti al di fuori della blockchain) con la funzionalità standard di scripting di Bitcoin, che realizza smart-contracts.

L'idea cruciale di TumbleBit è creare un servizio che risolve puzzle crittografici per Bitcoin, conosciuti come *RSA puzzles*. Essi sono difficili da rompere

così come lo è ricavare una chiave privata di RSA a partire dalla corrispondente chiave pubblica. Inoltre, viene usato il blinding di RSA che evita che sia collegata la soluzione ad un particolare puzzle. Si è solamente a conoscenza del fatto che la soluzione risolve un puzzle che è stato creato, ma non quale precisamente.

Alice, l'entita pagante, immette i suoi Bitcoin in un canale di pagamento con il TumbleBit hub. L'hub viene pagato solo quando presenta una soluzione al puzzle crittografico di Alice. Questo protocollo chiamato *Puzzle-Solver* fa in modo che il server non possa entrare in possesso dei Bitcoin finché non presenta una soluzione al puzzle e, nel momento in cui ne viene presentata una, Alice deve pagare.
Dall'altro lato Bob, l'entità che viene pagata, ha stipulato un accordo analogo con l'hub. Il Tumbler quindi accetta di pagare Bob solo nel momento in cui egli presenti una soluzione al puzzle RSA. Questo protocollo è noto come *Puzzle-Promise*.
Grazie all'utilizzo del blinding dei puzzle RSA con un numero sufficiente di puzzle esca, è possibile rompere il legame tra il risolutore del puzzle e il creatore. Il server TumbleBit sa che la soluzione è valida, ma non è in grado di determinare esattamente quale enigma sta risolvendo. In questo modo si offusca di fatto la relazione tra pagante e beneficiario.

Il protocollo ha un funzionamento ottimale quando molte parti si impegnano nell'utilizzo di questo servizio e, alla fine, tutte le transazioni singole vengo unite in un'unica grande transazione che paga i saldi ai beneficiari. Inoltre, TumbleBit richiede che gli importi delle transazioni siano uguali, per prevenire possibili analisi sul tracciamento degli importi stessi.

Conclusione

Il tempo ha dimostrato che Bitcoin è un sistema di pagamento decentralizzato molto sicuro ed efficiente, tuttavia la sua elevata trasparenza mette a rischio la privacy dei suoi utenti. Il problema risulta persino più rilevante pensando che una violazione sistematica della privacy può ridurre anche la fungibilità della valuta. Inoltre, ci si aspetta che vengano sviluppati attacchi sempre più sofisticati per la violazione della privacy. Pertanto, è necessario che lo sviluppo di tecniche per aumentare la sicurezza e la privacy vada di pari passo.

Si è notato che i servizi di mixing citati sono molto efficienti per un miglioramento della privacy. Tuttavia, le loro implementazioni hanno ancora funzionalità limitate. Sono tecniche recenti ed in costante ricerca, perciò lo spazio di miglioramento è vasto, come ad esempio il nuovo protocollo *WabiSabi* per il Wasabi Wallet, che permetterebbe di effettuare CoinJoin in maniera più accessibile e veloce.
Quando queste tecniche raggiungeranno un livello di sviluppo tale da permetterne l'utilizzo abituale, la privacy del sistema Bitcoin otterà una qualità mai vista prima nei sistemi di pagamento elettronici.

Riferimenti

[1] Chris Belcher. *CoinSwap GitHub*. URL: https://gist.github.com/chris-belcher/9144bd57a91c194e332fb5ca371d0964.

[2] *Binance - Coin mixing and CoinJoins*. URL: https://academy.binance.com/it/articles/coin-mixing-and-coinjoins-explained.

[3] *Bit2Me - What is CoinJoin*. URL: https://academy.bit2me.com/en/que-es-coinjoin/.

[4] *Bitcoin Wiki - CoinJoin*. URL: https://en.bitcoin.it/wiki/CoinJoin.

[5] *Bitcoin Wiki - JoinMarket*. URL: https://en.bitcoin.it/wiki/JoinMarket.

[6] *Bitcoin Wiki - PayJoin*. URL: https://en.bitcoin.it/wiki/PayJoin.

[7] *Bitcoin Wiki - Privacy*. URL: https://en.bitcoin.it/wiki/Privacy.

[8] *Bitcoin Wiki - TumbleBit*. URL: https://en.bitcoin.it/wiki/TumbleBit.

[9] *Boltzmann GitHub*. URL: https://github.com/Samourai-Wallet/boltzmann.

[10] Jordan Clifford. "TumbleBit: A new kind of mixing service on Bitcoin". In: (2019). URL: https://medium.com/scalar-capital/tumblebit-96b346f2e86b.

[11] Simin Ghesmati et al. *Bitcoin privacy - A Survey on Mixing Techniques*. URL: https://eprint.iacr.org/2021/629.pdf.

[12] Colin Harper. "Wasabi Wallet Is Revamping Its CoinJoin Design to Allow Bitcoin Mixing With Differing Values". In: (2020). URL: https://www.coindesk.com/wasabi-wallet-coinjoin-design.

[13] *JoinMarket GitHub*. URL: https://github.com/JoinMarket-Org/joinmarket-clientserver.

[14] LaurentMT. "Introducing Boltzmann". In: (2017). URL: https://medium.com/@laurentmt/introducing-boltzmann-85930984a159.

[15] *Vemprarua - What is CoinJoin*. URL: https://it.vemprarua.org/what-is-coinjoin-detailed-explanation.

[16] *WabiSabi GitHub*. URL: https://github.com/zkSNACKs/WabiSabi.

[17] *Wasabi docs - ZeroLink protocol*. URL: https://docs.wasabiwallet.io/using-wasabi/CoinJoin.html.

[18] *ZeroLink GitHub*. URL: https://github.com/nopara73/ZeroLink.

BLOCKCHAIN
E GIOCHI DIGITALI

Elena Pitino, Matteo Pappadà, Alessandro Pino, Luca Montaldo

Indice

1 Introduzione

L'ecosistema blockchain sta crescendo ad un ritmo accelerato e, mentre la tecnologia viene utilizzata principalmente in network di criptovalute, consente anche soluzioni innovative in una vasta gamma di settori: sanitario, logistico, IoT, beneficenza e altri. Negli ultimi anni una mini rivoluzione sta interessando anche il mondo dei videogiochi.

2 L'industria del gaming

L'industria dei videogiochi è in piena espansione e non conosce crisi, il comparto vanta all'attivo miliardi di giocatori in tutto il mondo e nel 2020 ha fatturato 159 miliardi di USD. Una compagnia di videogiochi conta i suoi guadagni dividendoli in due categorie: i Game revenue, ossia gli incassi generati dalla vendita del gioco, e le Game spend, vale a dire le microtransazioni fatte all'interno del videogioco. Quest'ultime sono fatte dagli utenti per acquistare add-on, crediti e sbloccare contenuti aggiuntivi. Infatti, giochi quali Fortnite, di tipo free-to-play, producono solo ricavi derivanti dalle vendite in-game. Questi acquisti permettono ai gamers di ottenere vantaggi, sbloccare contenuti e accedere a ricompense esclusive. All'interno di alcuni videogame è possibile "acquistare" con denaro vero una valuta virtuale da spendere nel gioco stesso.
Tuttavia, dato che il database appartiene a una singola compagnia, i giocatori non hanno una reale proprietà dei loro account e oggetti. Infatti, la maggior parte dei videogiochi online si basa su un modello centralizzato, in cui tutti i dati sono archiviati su un server interamente controllato dagli amministratori del gioco. Questi dati includono le informazioni dell'account e la storia del server, che registra e archivia tutti gli eventi e gli asset in-game ottenuti dai giocatori. In questo modo i giochi, poiché gestiti da server centralizzati, presentano molti limiti e vulnerabilità, che possono essere dovuti da:

- Malfunzionamento del server a causa di problemi tecnici
- Infiltrazione di hacker nel sistema
- Interruzione del gioco
- Mancanza di trasparenza sui meccanismi e i costi del gioco
- Manipolazione dell'economia interna da parte di sviluppatori e amministratori

In altre parole, il potere è nelle mani delle compagnie di videogiochi. Fortunatamente, però, la tecnologia blockchain è in grado di eliminare o mitigare molti di questi problemi.

3 I vantaggi della tecnologia Blockchain

Come database distribuito, un sistema basato su blockchain può essere usato per verificare e proteggere qualsiasi tipo di dati digitali, inclusa la storia in-game, gli oggetti digitali e gli asset tokenizzati. L'idea centrale è quella di togliere il potere dalle compagnie di videogiochi e restituirlo ai giocatori. In questo modo, ogni giocatore può avere il pieno controllo sui propri account e asset digitali, ed è libero di scambiare questi asset in qualsiasi momento. Infatti, la tecnologia blockchain è un vantaggio per molti settori carenti in termini di affidabilità. Al centro di questa tecnologia c'è l'abilità di creare un ambiente trustless (che non necessità di fiducia reciproca) che facilita transazioni immutabili tra due sconosciuti su internet.
La blockchain fornisce essenzialmente un database trasparente e decentralizzato che è crittograficamente archiviato su vari nodi sparsi nel mondo. Nessuna entità singola controlla la rete, quindi non esiste un singolo database da attaccare e non c'è la possibilità di cancellare una transazione una volta effettuata. Il basso costo delle transazioni rende semplice trasferire monete o token in tutto il mondo. Queste transazioni non passano attraverso nessun *man in the middle*, il che le rende più sicure e quasi in tempo reale. La blockchain, quindi, risolve alcuni dei problemi dell'industria odierna dei giochi digitali e presenta numerosi vantaggi che possono rivoluzionare il mondo del gaming.

3.1 I vantaggi per i giocatori

Verificabilità e trasparenza

Una novità dei giochi moderni è l'utilizzo di asset o risorse per completare missioni. Si può avere bisogno di armi, attrezzi, ambienti, auto, aerei, personaggi e opere d'arte. I giochi moderni sono dipendenti da questi asset, piuttosto rari all'interno del videogame, che possono essere o acquistati dai giocatori o guadagnati tramite l'avanzamento nel gioco. Tuttavia, al momento, non sono garantite l'affidabilità e la trasparenza. Questi asset sono virtuali, per cui gli sviluppatori del gioco potrebbero produrne una quantità illimitata o manipolare il mercato fornendo alcuni asset solo a determinati giocatori. Sorge quindi la necessità di trasparenza e verificabilità.
La blockchain permette la tokenizzazione di questi asset e la creazione di mercati decentralizzati, in modo da poter comprare asset virtuali ad un prezzo onesto e trasparente, basato su un registro pubblico e aperto, quindi consultabile da tutti.

Scarsità verificabile

Una delle proprietà che fa sì che questi asset abbiano un valore è la loro scarsità. Tuttavia, con la situazione attuale è impossibile per un giocatore conoscere la scarsità di un particolare oggetto.
Se questi asset vengono registrati su una blockchain, i giocatori possono verifi-

care facilmente la quantità totale di ciascuno di essi sul registro pubblico. Ciò incrementa la fiducia generale e quindi il valore del mercato stesso.

Sicurezza

Le piattaforme di gaming si appoggiano su server centralizzati e le transazioni sono spesso fatte da smartphone o pc senza misure di sicurezza adeguate. In più, gli asset posseduti nel gioco sono soggetti a furti. Non sono sicuri come gli account bancari, nonostante il valore che possono avere alcuni account di gaming.
La blockchain è conosciuta per essere il modo più sicuro per depositare valore ed è stata creata per essere teoricamente inattaccabile. Accumulare asset di gioco digitali su una blockchain accresce la sicurezza di un giocatore che ha faticato e speso tempo per collezionarli.

Oggetti collezionabili

Molti giochi esistenti adottano una propria valuta di gioco che non permette di trasferire il suo valore su altre piattaforme, mentre un gioco basato su blockchain può sfruttare criptovalute (token fungibili) come meccanismo di pagamento multipiattaforma. Inoltre, possono essere utilizzati i token non fungibili (NFT) per identificare in modo univoco gli asset o le risorse di un gioco (ad esempio una spada rara vinta sconfiggendo un particolare boss o la reputazione di un giocatore o altri asset che sono unici e collezionabili). Gli NFTs possono essere usati per rappresentare questi oggetti e far sì che siano facilmente conservabili su un wallet, meno costosi da vendere e scambiare su un mercato aperto. Se molti giochi diversi utilizzano lo stesso standard NFT per i loro asset, questi avranno un valore di rivendita al di fuori del gioco e potranno essere scambiati sui vari mercati. Siccome la blockchain può dimostrare che i giocatori possiedono veramente determinati beni, la community di un gioco potrebbe essere disposta a pagare somme sostanziali per determinati oggetti.



Asset Name (Game)	Dragon (CryptoKitties)	Hyperion (Gods Unchained)	1-1-1 (F1 DeltaTime)
Purchase Price	600 ETH (US $172K)	146 ETH (US $61K)	416 ETH (US $110K)
Purchase Date	9/4/2018	8/7/2018	5/30/2019
Item Description	"Gen 0" kitties are the rarest of cryptokitties (capped at 50k supply)	One of the four unique Gods Unchained Genesis Titans, and the only one available for purchase	The first F1 virtual car NFT released by F1 and Animoca Brands' new collaboration

Figura 1: Esempi di somme pagate per alcuni oggetti digitali

Scambio di asset digitali su exchange decentralizzati

Al giorno d'oggi, gli asset digitali sono scambiati all'interno del gioco o su exchange come *Wax*, *OpenSea* e *RareBits*. Questi scambi potrebbero essere ancora più trasparenti su un exchange decentralizzato in forma tokenizzata. Infatti, acquistando oggetti su exchange come quelli citati, si ha sempre il rischio di comprare asset falsi o di venire truffati. Gli exchange decentralizzati su blockchain risolvono questo problema.

Tempo e costo di una transazione

Il gaming al giorno d'oggi ha una portata mondiale. Tantissime persone, anche da paesi differenti, giocano tra di loro agli stessi giochi. Come possono i giocatori trasferire i loro asset senza che passino dei giorni per processare pagamenti e senza il bisogno di districarsi tra problemi legali?
La blockchain può abilitare pagamenti istantanei in tutto il mondo, ciò significa anche che non ci sarebbe alcuna restrizione.

3.2 I vantaggi per gli sviluppatori

Sebbene sia certamente importante migliorare l'esperienza dei giocatori, è anche necessario comprendere i vantaggi che gli sviluppatori hanno incorporando la blockchain nello sviluppo dei loro giochi.

Mercato emergente

Per ogni grande successo indie, ci sono decine, centinaia o migliaia di giochi alternativi che non decollano mai. Forse il più grande ostacolo da superare per uno sviluppatore o uno studio indipendente è la sua capacità di distinguersi e attirare l'attenzione dei giocatori al fine di attirare un pubblico e creare una comunità attorno al loro gioco. Molti sviluppatori di giochi tradizionali si stanno cimentando in progetti di gioco blockchain, ciò conferma l'assunto che ci sarà un gioco blockchain rivoluzionario, e c'è una gara in corso per essere coloro che lo svilupperanno. I primi grandi giochi blockchain che cattureranno l'attenzione a livello mondiale saranno dei successi enormi e i pionieri che li avranno sviluppati saranno ben ricompensati.

Riduzione dei costi

Per un settore che comporta ricavi limitati come lo sviluppo di giochi, i costi associati possono essere paralizzanti. Anche quando i giochi hanno una community di giocatori attiva, l'onere della pubblicazione, della manutenzione del server, della moderazione dei giocatori e così via possono essere insostenibili per i giochi che non "monetizzano adeguatamente" i loro giocatori. Costruendo giochi su blockchain, tutti questi costi possono essere potenzialmente esternalizzati ai miner o ai validatori responsabili della propagazione della rete. La

pubblicazione e la propagazione dei giochi online possono potenzialmente essere un'attività a costo zero.

Nuovi generi di sviluppo

Come detto in precedenza, gli sviluppatori e gli studi di sviluppo, in particolare nel mondo dell'indie, sono sempre alla ricerca di modi per differenziarsi. I giochi blockchain hanno delle caratteristiche e peculiarità uniche che possono essere sfruttate nella formazione di generi di gioco completamente nuovi e precedentemente sconosciuti. Ad esempio esistono già dei giochi di tipo "play-to-earn", in cui il gameplay ruota attorno al concetto di guadagnare dei profitti nel mondo reale semplicemente giocando.

Creazione di valore

Tante tendenze recenti nel mondo del gaming si sono focalizzate sul sottrarre valore ai giocatori attraverso i modelli "freemium" e "pay-to-win" che incentivano la dipendenza e la spesa di denaro rispetto al vero divertimento. Per varie buone ragioni, molti sviluppatori si oppongono a questo modello di profitto e alle implicazioni parassitarie che ha sull'industria nel suo insieme. Con la blockchain, gli sviluppatori hanno più sbocchi creativi per guadagnare profitti. I giochi blockchain consentono agli sviluppatori di creare e forgiare risorse crittografiche con valore reale, trasferibile e intrinseco. Questo non solo offre ai giocatori una maggiore capacità di profitto, ma offre anche agli sviluppatori meccanismi di profitto meno invadenti.

Playerbase migliorata

I giocatori esistenti e futuri che costituiscono il gaming su blockchain sono un sottoinsieme interessante della popolazione dei consumatori. Questi giocatori hanno un elevato interesse per il loro gameplay e una maggiore attenzione nella ricerca del profitto attraverso il loro gioco. Il gaming su blockchain collega gli sviluppatori a comunità di "super giocatori", che investono e impiegano molte più risorse sui loro giochi. È possibile creare un rapporto più attivo, profondo e vantaggioso tra le due parti, a differenza di quanto sia possibile nei giochi tradizionali.

4 Come funziona

4.1 Architettura dei giochi su Blockchain

La figura successiva illustra l'architettura utilizzata dai giochi su blockchain. Diversamente rispetto ai giochi tradizionali, un nuovo giocatore registra un address sulla corrispondente blockchain prima di iniziare la sua sessione di gioco. Questo address, a cui si accede da un wallet, servirà come destinazione per gli asset virtuali del giocatore corrispondente. Grazie a questo meccanismo, il server

del gioco delega alcune funzioni alla blockchain sotto forma di smart contract, che gestiscono gli asset virtuali del giocatore. Una blockchain che supporta gio-

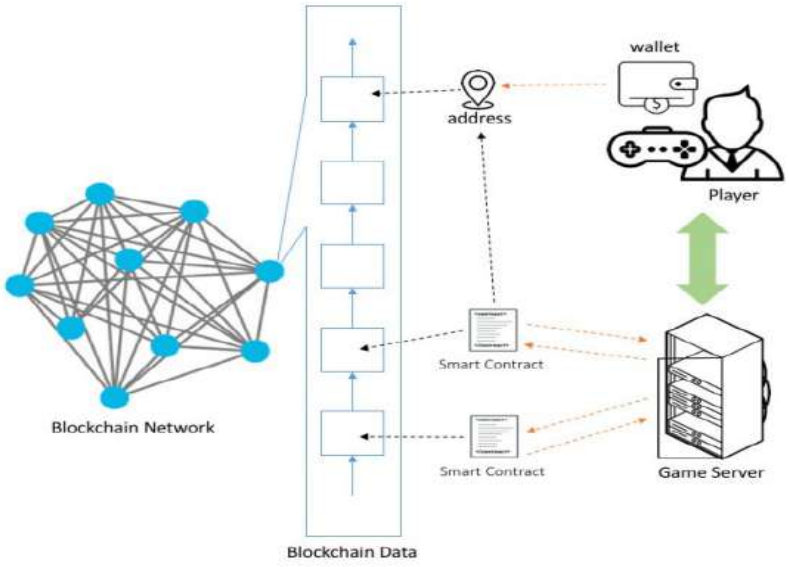


Figura 2: Architettura di un gioco su blockchain

chi digitali deve fronteggiare molte transazioni al secondo e quindi performare molto più rapidamente. Le blockchain tradizionali che adottano modelli di consenso PoW, come Ethereum, sono state spesso inadatte nel soddisfare questi bisogni. Quindi, nuovi modelli di consenso sono stati proposti da nuove piattaforme come EOS, Tron, Neo, Qtum, Nebulas, ecc.
Al momento la maggior parte dei giochi digitali sfruttano la blockchain di Ethereum. Nel settore di oggi, Ethereum è la blockchain più popolare in termini di sviluppo di giochi. DappRadar riporta che ci sono oltre 400 giochi blockchain registrati su Ethereum e centinaia di altri in diverse fasi di sviluppo. Però l'utilizzo di questa blockchain ha portato alla luce diversi problemi, tra gli altri i principali sono un alto costo delle tasse di transazione e alcuni problemi di scalabilità, infatti la blockchain risulta avere problemi quando un gioco nuovo viene utilizzato in massa.
Per risolvere questi problemi stanno iniziando a nascere nuovi progetti di blockchain il cui scopo è proprio quello dell'implementazione di giochi digitali.
Una delle più famose si chiama Enjin, il cui scopo, oltre a fornire caratteristiche analizzate in precedenza (verificabilità, trasparenza, sicurezza,...) promette di risolvere i problemi che si sono riscontrati sulla blockchain di Ethereum.

4.2 Enjin

Enjin è una società che fornisce un ecosistema di prodotti di gioco interconnessi basati su blockchain. L'azienda è nota anche per aver creato nel 2018 il token 1155, che ridimensiona uno dei più grandi problemi riguardanti l'utilizzo della tecnologia blockchain nei videogiochi, ovvero la scalabilità.
Grazie alla nascita di Ethereum è stato possibile introdurre nelle blockchain il concetto di "smart contract", generati attraverso una macchina di Turing virtuale. La maggior parte degli smart contract tuttora esistenti sono per lo più token ERC-20 che rappresentano dei veri e propri asset virtuali. Una delle caratteristiche chiave di questi token è la loro fungibilità (fungible token).
Nei giochi però, oltre alla creazione di una valuta digitale per l'economia interna, vi è la necessità di associare tutta una serie di oggetti alla blockchain. Questi oggetti diventano in questo modo unici e realmente di proprietà degli utenti, che possono gestirli tramite il proprio wallet. Questo concetto di collezionabilità ha portato Witek Radomski, cofondatore di Enjin, a creare nel 2017 il primo protocollo di base per token non fungibili. Da questo protocollo è stato ricavato il modello per i token ERC-721.
Pertanto, per ogni asset in gioco si assegna un token ERC-721? In realtà un simile approccio non sarebbe praticabile per un gioco come World of Warcraft, dove esistono svariate migliaia di oggetti differenti. Se si creassero centinaia di migliaia di ERC-721 per ogni singolo item, la blockchain Ethereum si intaserebbe. È per questo che sono stati creati gli ERC-1155, che hanno un approccio differente rispetto ai predecessori: gli oggetti vengono registrati tutti in un unico contratto, con la possibilità di riconoscere agevolmente i vari tipi di oggetti. Nello stato del contratto è possibile gestire gli oggetti grazie ai "Token ID" e sono presenti tutte le condizioni necessarie per gestire la collezionabilità. Quello che fa il protocollo ERC-1155 è consentire il rilascio di token sia fungibili che non fungibili: una funzionalità fondamentale per ogni sviluppatore di videogiochi. The Sandbox, uno dei giochi basati su Ethereum più attesi del momento, ha implementato alcuni ERC-1155.

Figura 3: Logo Enjin

Enjin è un ecosistema dedicato ai creatori di videogiochi, che consente loro di raccogliere fondi per lo sviluppo: sia prima del lancio, grazie ad elementi in prevendita, che quando il gioco è già attivo e funzionante, tramite fee addebitate agli utenti quando gli oggetti in-game vengono trasferiti verso marketplace

esterni. Questa è una grande innovazione per gli sviluppatori: anche su WoW i giocatori scambiano gli oggetti su marketplace esterni, ma in questo caso l'unico a ottenere un guadagno è l'intermediario di terze parti. Con Enjin, tali fee vanno direttamente agli sviluppatori.

5 Le origini del gaming su Blockchain

Prima di esporre la nascita e l'evoluzione del gaming su blockchain, diamo qualche nozione di base e facciamo qualche precisazione sui termini utilizzati in questo capitolo.

Cripto Giochi vs Giochi Blockchain

Chiamiamo "cripto gioco" qualsiasi videogioco che, in qualche modo, incorpora una criptovaluta. Invece, indichiamo con "gioco blockchain" qualsiasi gioco in cui parte o tutto il gameplay si svolge direttamente su una blockchain. In altre parole, gli input o l'attività di gioco in un gioco blockchain consistono in transazioni trasmesse dai giocatori sulla rete. In un "vero" gioco blockchain, l'intero gioco persiste sulla catena. Ciò significa che il mondo di gioco gira, senza server, sulla rete stessa. In generale, ogni gioco blockchain è anche un cripto gioco, ma non tutti i cripto giochi sono giochi blockchain.

Applicazioni decentralizzate o DApp

Le applicazioni decentralizzate, o DApp, si riferiscono a progetti rivolti agli utenti con cui si interagisce tramite reti blockchain. Molti cripto giochi esistono come DApp su varie blockchain, mentre praticamente tutti i giochi blockchain possono essere rappresentati come DApp.

5.1 Cripto giochi

È difficile individuare esattamente il primissimo cripto gioco, ma già nel 2011 su alcuni forum si discuteva su come sviluppare dei giochi legati a Bitcoin. Tuttavia, molte delle idee e dei giochi proposti non sono mai andati a buon fine al di là della loro discussione sui forum. Nel 2013, ci sono stati diversi casi di giochi a tutti gli effetti che operavano tramite Bitcoin. C'è stata un'ondata simile nel 2015, ma questa ondata successiva si è espansa incorporando varie altcoin.

Giochi d'azzardo e Giochi PvP con Bitcoin

Nel 2013 l'utilizzo di Bitcoin nel gioco d'azzardo era già in atto e fiorente. Seals with Clubs era una poker room online con diverse centinaia di giocatori costantemente attivi. Primedice, un altro gioco d'azzardo, era già in funzione da un po' di tempo e stava espandendo la sua popolarità a centinaia di migliaia di

utenti. Il gioco PvP, ovvero player vs player, tramite Bitcoin ha quindi rappresentato il passo avanti più logico. Piuttosto che giocare contro il banco, questi giochi PvP offrivano dei giochi semplici che permettevano ai giocatori di competere l'uno contro l'altro.
Tra i primi giochi PvP, il più popolare e di successo è stato un sito web chiamato Gambit (ora Cryptogames). Lanciato alla fine del 2013, Gambit offriva una serie di giochi da tavolo e di carte attraverso i quali i giocatori gareggiavano e scommettevano Bitcoin l'uno contro l'altro. Questi giochi includevano una serie di cripto-spinoff di giochi classici come Monopoly, Battaglia navale e Risiko. Ciò che rendeva Gambit molto attraente per i giocatori erano le generose offerte del faucet. Ogni gioco consentiva agli utenti di giocare senza buy-in e ottenere vincite modeste. Il gioco Monopoly ha avuto le vincite più alte, anche perché non c'erano limiti o restrizioni sulla frequenza con cui qualcuno poteva accedere al faucet.

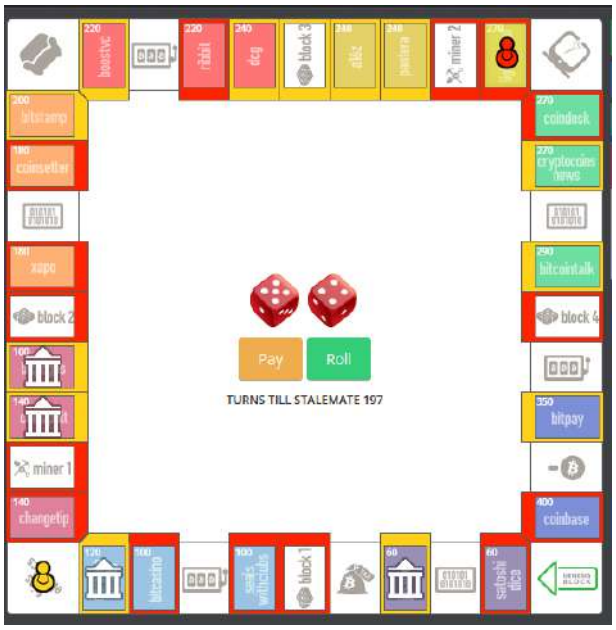


Figura 4: Il gioco Monopoly su Gambit

Sfortunatamente, il faucet di Gambit era troppo generoso: i giocatori non si preoccupavano del rischio di scommettere la propria moneta e, inoltre, si limitavano a giocare ai giochi gratuiti nella speranza di moltiplicare le vincite abbastanza volte per poi incassare. Nel tempo il sito ha ridotto i pagamenti del faucet e ha incorporato nuove funzionalità, come ad esempio le classifiche mensili dei giocatori migliori . Tuttavia, non è riuscito ad attirare a sufficienza l'attenzione per essere considerato qualcosa di più di un sito di giochi faucet BTC.

Altri giochi PvP hanno subito destini simili. Il loro successo si basava sul fascino dei guadagni del faucet, ma era un equilibrio delicato da mantenere: troppo poco, nessuno gioca; troppo, e lo sviluppatore del gioco finiva in perdita. Altri esempi includevano Bombermine, uno spin-off online di Bomberman in cui i giocatori scommettevano Bitcoin, e Gamerholic, un sito web competitivo composto da giochi arcade retrò. C'erano anche siti web che fungevano da piattaforma di scommesse peer-to-peer per giochi tradizionali come League of Legends e Counter-Strike: Global Offensive. Tuttavia, tutti questi progetti non sono riusciti a ottenere una popolarità significativa.

Server di Minecraft

Il successivo ramo di sviluppo oltre ai giochi PvP è arrivato attraverso un mezzo interessante: Minecraft. Il famoso gioco a blocchi ospitava server specializzati che incorporavano Bitcoin già alla fine del 2012. Gli sviluppatori hanno creato dei plug-in open source che consentivano agli avatar in gioco di avere dei wallet Bitcoin, mentre diversi proprietari di server indipendenti hanno creato dei server personalizzati attorno a questa funzionalità.
Il primo server Bitcoin di Minecraft documentato è MinecraftCC. Su MinecraftCC, i giocatori ricevevano pagamenti settimanali in Bitcoin per ogni azione sul server, come ad esempio posizionare blocchi, uccidere mostri e costruire strutture. Gli amministratori hanno trovato modi creativi per finanziare i pagamenti, ad esempio tramite annunci pubblicitari a pagamento messi all'asta sui forum Bitcointalk. Tuttavia, entro il 2016, il programma è diventato insostenibile e i guadagni in Bitcoin sono stati rimossi. Oggi il server è ancora attivo, ma il ruolo che ricopre ancora la criptovaluta non è chiaro.
Un altro esempio interessante è BitQuest, che è stato lanciato più tardi, ma funziona ancora oggi. BitQuest aveva un nuovo approccio per convertire l'hardware del server in un miner Bitcoin che trasformava i guadagni legati al mining in beni di gioco che potevano essere raccolti uccidendo i mob, ovvero le creature presenti nel gioco. Tuttavia, poiché col passare del tempo il mining è diventato molto più competitivo, ora non si guadagnano più BTC giocando.

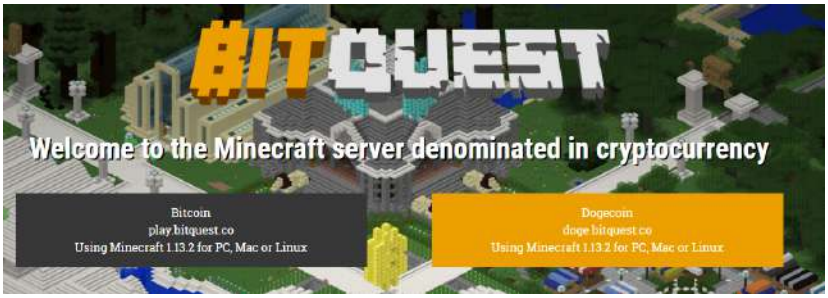


Figura 5: Schermata iniziale di BitQuest

Ci sono stati altri server che hanno incorporato diverse altcoin negli anni successivi. In particolare, ci sono stati tentativi di creare server specializzati in esecuzione su Dogecoin e DigiByte.

Giochi Indie

Delle tre categorie di cripto giochi, i giochi indie (abbreviazione dell'inglese independent) rappresentano la più emozionante e rivoluzionaria delle prime applicazioni blockchain. Con vari livelli di successo, diversi individui e team hanno sviluppato dei giochi da zero che funzionavano tramite Bitcoin e altre altcoin.
Per molti aspetti, Dragon's Tale è stato il gioco indie di maggior successo in esecuzione su Bitcoin. Con radici che risalgono al 2011, precede anche gli altri giochi elencati. Dragon's Tale era un gioco MMO (Massively Multiplayer Online) con qualche caratteristica di un RPG (Role Playing Game) il cui gameplay era incentrato sul gioco d'azzardo. Era composto da numerosi giochi di varia fortuna e abilità su cui scommettere BTC, insieme a semplici quesiti ed esplorazioni per guadagnare BTC. Dragon's Tale ha iniziato a svanire dopo diversi anni e l'attività è sembrata sostanzialmente scemare nel 2015-2016.
Altri esempi includono Hammercoin, Island Forge e Bitfantasy. Tutti questi giochi sono svaniti a causa della crescente inadeguatezza di Bitcoin come valuta di gioco: i tempi di transazione lenti e le commissioni di transazione sempre più costose hanno reso impossibile la crescita di qualunque gioco.
C'è stato più successo tra i primi cripto giochi che hanno adottato l'approccio più moderno di introdurre le proprie risorse all'interno del gameplay. L'esempio più famoso e il primo di grande successo è Spells of Genesis. Spells of Genesis è un gioco di carte collezionabili del tipo sparabolle. L'economia del gioco è basata sulle monete (asset fungibili) di Counterparty (XCP) chiamate BitCrystals (BCY), mentre le carte rappresentano asset XCP non fungibili. Spells of Genesis ha visto una crescita impressionante in termini di attività su Google Play Store, come anche la valutazione dei suoi asset. Dopo un lancio soft nel 2016 e il lancio principale nel 2017, SoG ha accumulato oltre 10.000 di download e BitCrystals è cresciuta fino a raggiungere una capitalizzazione di mercato a otto cifre per la maggior parte del 2016 e del 2017.
In un modo simile si è diffuso Nexium, un MOBA (Multiplayer Online Battle Arena) ambientato nello spazio in cui i giocatori gareggiavano tra loro e scommettevano NXM sugli esiti delle battaglie tra astronavi. Nexium e NXM hanno ricevuto attenzioni e valutazioni simili a SoG e BCY per tutto il 2017, ma hanno poi perso la loro popolarità.

Monete da gioco

Anche prima dell'ICO-mania (ICO sta per Initial Coin Offering) del 2017, c'erano un certo numero di altcoin costruite per essere monete da gioco universali, i predecessori di progetti come Enjin Coin. Queste altcoin includevano Hyper, GoldPieces (GP), GameCredits (GAME), DigiByte (DGB) e molte altre. Ciascuno di questi progetti ha adottato approcci community-oriented per incen-

tivare nuovi giochi e sviluppatori ad adottare le loro valute.
Hyper, GP e DGB offrivano tutte, prima o poi, meccanismi in cui gli utenti potevano guadagnare ciascuna moneta attraverso il gameplay su giochi popolari come Team Fortress 2 (Hyper e GP) e League of Legends (DGB). GameCredits ha adottato un approccio diverso, concentrando invece l'attenzione sulla creazione di nuovi giochi appositamente per l'uso di GAME.

5.2 Giochi Blockchain

Mentre i primi cripto giochi hanno sicuramente svolto un ruolo utile gettando le basi per i progetti futuri, gli sviluppi più ambiziosi e impressionanti in questi anni sono rappresentati dai primi giochi blockchain. Siccome non c'era alcun pattern su come si dovesse collocare esattamente un gioco su una blockchain, i primi giochi di questo tipo sono molto meno numerosi.
I due più importanti di questa classe sono stati Huntercoin e Motocoin. Sebbene fossero molto diversi tra loro e vedessero la propria serie di successi e mancanze, entrambi sono di ispirazione per i giochi blockchain di oggi.

Huntercoin

Huntercoin è un gioco basato sulla blockchain omonima creata appositamente a questo scopo, la cui rete promuove un mondo virtuale in cui i giocatori fanno combattere tra di loro i propri personaggi per difendere e raccogliere monete, ovvero token HUC.
La rete principale di Huntercoin (così come il mondo di gioco) è stata lanciata nel febbraio 2014. I giocatori interagivano con il mondo di gioco diventando un nodo della blockchain Huntercoin e utilizzavano le transazioni per creare e dirigere i loro personaggi con l'intento di raccogliere HUC mentre essi venivano generati all'interno della mappa. Huntercoin è stato "merge-mined" verso Bitcoin, ovvero una modalità di mining che prevede l'utilizzo di due blockchain parallele: una parent, (in questo caso BTC) e una child (in questo caso HUC). Con questo meccanismo, se si crea un blocco nel parent, si riceve la ricompensa per entrambe le chain, mentre se si inserisce un blocco nella child, si riceve la ricompensa solo per quest'ultima. I miner ricevevano il 10% dei guadagni dovuti all'approvazione del blocco HUC, mentre il restante 90% andava a finire sulla mappa per essere raccolto dai giocatori, modalità anche detta "human mining".
Huntercoin è stato inizialmente progettato come un esperimento con una durata di un anno, per testare la praticità e la longevità di tale impresa. Tuttavia, è stato un successo così travolgente che la community ha spinto una fork per eliminare il limite temporale. Questo gioco è stato un banco di prova di enorme successo per il funzionamento dei giochi su blockchain, ma è stato ancora più rivoluzionario per la sua modalità di creazione del valore da riservare ai giocatori. Quest'ultimi hanno guadagnato collettivamente oltre 1 milione di HUC, convertiti in Bitcoin su Poloniex (un exchange di cripto-asset). La blockchain di Huntercoin ha generato milioni di dollari di profitti per i giocatori semplicemente attraverso la propria propagazione e creazione di HUC. Questa è la base

per il modello "play-to-earn", con la quale gli sviluppatori di giochi blockchain stanno tentando di creare giochi strutturati in modo tale che i loro giocatori guadagnino profitti nel mondo reale tramite le loro attività. Huntercoin è stato rimosso dagli exchange all'inizio del 2019, ma continua ad avere utenti attivi anche al giorno d'oggi.

Figura 6: Gameplay di Huntercoin

Motocoin

Il 20 maggio 2014, poco dopo il lancio di Huntercoin, è stato lanciato Motocoin con il suo innovativo meccanismo di consenso "proof-of-play" (prova di gioco). Analogamente a Huntercoin, gli utenti si sono uniti al gioco come nodi della rete, ma in Motocoin il gameplay stesso era il meccanismo per ottenere le ricompense legate alla creazione di blocchi.
Motocoin era un gioco di corse motociclistiche in un ambiente generato proceduralmente con ogni blocco. Il primo giocatore a validare il blocco correndo verso il traguardo guadagnava la ricompensa MOTO del blocco. Ad ogni nuovo blocco, i personaggi si resettavano e la mappa veniva sostituita con il suo successore generato proceduralmente.
Motocoin ha riscontrato un'attività considerevole per vari anni successivi al lancio, tuttavia è stato paralizzato dal fatto che il gioco era dominato dai bot. Il percorso più veloce verso il traguardo era una soluzione facile da calcolare ed eseguire per i bot, che quindi riuscivano a raggiungere il blocco successivo più velocemente di quanto potesse fare qualsiasi giocatore manualmente, e così il gameplay di MOTO è stato soffocato. Ufficialmente, Motocoin ha ancora una capitalizzazione di mercato di circa 230.000 EUR, ma la moneta non ha visto alcuna attività di trading significativa per la maggior parte della sua vita.

Figura 7: Gameplay di Motocoin

6 I giochi più famosi

L'attuale diffusione dei giochi blockchain rappresenta un'impennata iniziata nel dicembre 2017. Se la prima generazione di giochi può essere denominata come "generazione zero", allora il campo dei giochi blockchain e cripto giochi, da dicembre 2017 ad oggi, indica una prossima generazione.

6.1 CryptoKitties

CryptoKitties è nato con lo scopo di rendere accessibile la tecnologia blockchain al consumatore medio, attraverso funzionalità di gioco che sfruttano le applicazioni uniche della blockchain, e una piattaforma adatta a tutti i livelli di conoscenza tecnica.
CryptoKitties è un gioco cripto-collezionabile su Ethereum in cui gli utenti allevano e raccolgono gatti unici rappresentati come token non fungibili ERC-721. L'intero ambito del gioco ruota intorno all'acquisto e alla vendita all'asta di gatti da scambiare e allevare. Ogni gattino ha un genoma unico che definisce il suo aspetto e le sue caratteristiche. Capire come farli accoppiare per ottenere nuovi attributi rari è alla base del gioco. Più un gatto è raro, più ovviamente è alto il suo valore. Alcuni gatti rari possono arrivare a costare migliaia di Euro. Ufficialmente, CryptoKitties è stato lanciato il 28 novembre 2017. Il 2 dicembre 2017, il primo CryptoKitty, giustamente chiamato "Genesis", è stato venduto all'asta per oltre 100.000 USD. Questa vendita ha battuto i record globali per gli oggetti da collezione digitali e ha attirato l'attenzione del pubblico in tutto il mondo. Ciò che è seguito alla vendita del 2 dicembre 2017 è stata una vera e propria corsa all'oro da parte delle persone che hanno inondato la rete Ethereum per allevare e vendere gatti virtuali.
Questa ondata di attività ha inferto un duro colpo alla rete Ethereum, che non era attrezzata per gestire i giochi su larga scala: il numero di transazioni non

confermate è aumentato di sei volte rispetto alle dimensioni precedenti e le commissioni di transazione sono aumentate in modo esponenziale. Gran parte di questo era dovuto al fatto che i giocatori di CryptoKitties hanno offerto enormi commissioni, e sono stati in grado di registrare le loro transazioni più facilmente e velocemente rispetto agli utenti "normali" di Ethereum. Ciò non solo ha causato un drastico rallentamento della rete, ma ha reso quasi impossibile l'attività al di fuori dell'allevamento di gatti virtuali.
Quando l'hype alla fine si è placato e le cose sono tornate più o meno alla normalità, l'introduzione di CryptoKitties ha fornito la prova che la rete Ethereum non era adatta a gestire dei giochi. Solidity è un linguaggio per contratti costruito per il trasferimento sicuro di grandi quantità di dati. I giochi, invece, richiedono una transazione rapida di piccoli pacchetti di informazioni. Pertanto, i giochi che hanno seguito CryptoKitties hanno ritenuto che la soluzione migliore per incorporare oggetti collezionabili ad alto costo simili ai gatti virtuali fosse spendere una quantità significativa di commissioni sulle transazioni per rendere fattibile ai giocatori il gameplay.
Anche CryptoKitties ha dato vita ad alcuni spunti di riflessione molto utili. La cosa più interessante è che CryptoKitties fa ora parte del più ampio "KittyVerse", una raccolta di oltre 20 DApp create da sviluppatori indipendenti che utilizzano i gattini ERC-721 come risorse per questi giochi. Ciò evidenzia una metrica fondamentale del gioco blockchain: questo tipo di gameplay può essere reso fluido e funzionale attraverso ambienti virtuali e mezzi di sviluppo differenti.

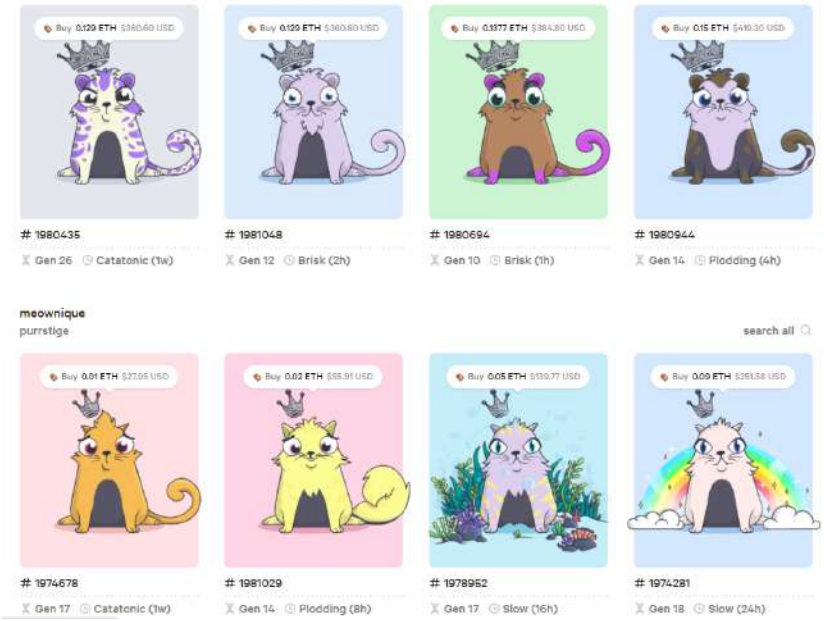


Figura 8: Alcuni CryptoKitties

6.2 Decentraland

Decentraland mette a disposizione una piattaforma di realtà virtuale basata sulla blockchain di Ethereum in cui non esistono letteralmente limiti: all'interno della piattaforma, gli utenti possono creare, sperimentare e monetizzare i propri contenuti e applicazioni.
Lo spazio virtuale 3D finito e attraversabile all'interno di Decentraland si chiama land, un asset digitale non fungibile mantenuto in uno smart contract Ethereum. Il terreno è suddiviso in parcelle identificate da coordinate cartesiane (x, y). Questi terreni sono permanentemente di proprietà dei membri della comunità e vengono acquistati utilizzando MANA, il token di criptovaluta di Decentraland. Tutto ciò è possibile, come già detto, tramite l'utilizzo della tecnologia blockchain, che dimostra univocamente la proprietà di un lotto.
Comprare land è semplice, tanto quanto usare MANA. Le piastrelle di terra misurano 10 metri quadrati. Non ci sono limiti sulla costruzione verticale, gli unici vincoli riguardano la base degli edifici. È importante notare che la terra di Decentraland è scarsa. Appositamente e volutamente progettata non solo per aumentarne la domanda, ma anche per migliorare l'esperienza utente complessiva e la capacità di scoprirne il contenuto. Ciò offre agli utenti il pieno controllo degli ambienti e delle applicazioni che essi creano, che possono variare da scene 3D statiche, ad applicazioni o giochi più interattivi. Si è liberi di costruire ciò che si vuole.

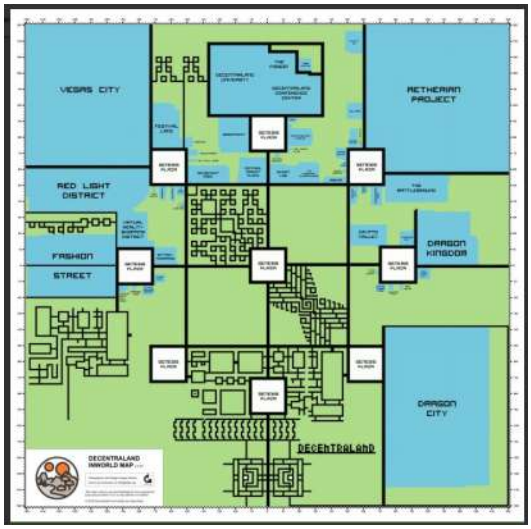


Figura 9: La mappa di Decentraland

Alcune delle opzioni suggerite dal team di Decentraland includono spettacoli di musica dal vivo, casinò, shopping, attività commerciali ed esposizioni. Tutto ciò si svolge in un mondo virtuale con una vista a 360 gradi capace di coinvolgere gli utenti attraverso il browser web o grazie all'utilizzo di un visore VR.

La più grande differenza tra Decentraland e le piattaforme VR esistenti è la proprietà. Il team che sta dietro al progetto crede che i mondi virtuali pubblici debbano essere gestiti e governati con degli standard aperti. Quindi nessuna organizzazione centrale deve imporre il proprio volere. Oltre ad essere di proprietà degli utenti, Decentraland consente loro di controllare completamente il lotto di terra che possiedono. Inoltre, i proprietari stessi sono in grado di ricavare dei guadagni derivanti dal valore generato dagli altri utenti. Completamente diverso dai sistemi in cui l'organizzazione centrale, che gestisce la piattaforma, prende una percentuale su ogni transazione. Senza un punto di centralizzazione, non vi è alcun gruppo in grado di decidere commissioni o tasse da pagare.

Figura 10: Una schermata di benvenuto su Decentraland

Il progetto Decentraland è iniziato nel giugno 2015 in quella che chiamano l'Età della pietra, quando la terra è stata modellata sotto forma di una semplice griglia e i pixel sono stati assegnati agli utenti tramite un algoritmo di Proof of Work. A marzo 2017, il progetto è entrato nella Età del bronzo, ed è stata aggiunta la vista in 3D. Coloro che hanno acquistato MANA, il token ERC20, sono stati in grado di ottenere lotti di terra e quindi di interagire con altri possessori della stessa criptovaluta. L'Età del ferro aggiunge il supporto multiplayer, insieme alla live chat e agli avatar. L'ultima fase, la Silicon Age, apre il mondo al supporto VR completo.
A marzo 2018 il team ha lanciato il Marketplace di Decentraland. Un vero e proprio mercato decentralizzato in cui acquistare e vendere tutte le proprie risorse Decentraland e on-chain. La vista Atlas, per esempio, offre una prospettiva dall'alto di ogni lotto, proprietà, strada, distretto e piazza di Decentraland. Per qualsiasi pacchetto attualmente in vendita nel Marketplace viene visualizzato lo stato, le coordinate e l'indirizzo pubblico del proprietario (se ha un proprietario).
Nell'agosto del 2017, c'è stata una ICO da 24 milioni di dollari per MANA, che è andata esaurita in pochi secondi. Inizialmente, ci sono state alcune critiche, poiché solo in pochi hanno acquistato la maggior parte dei token disponibili tramite l'ICO. L'asta è stata effettuata in modo decentralizzato e il MANA as-

segnato per le offerte vincenti è stato completamente bruciato. Quando i token vengono "bruciati", vengono inviati in un luogo inaccessibile e, a tutti gli effetti, fuori dalla portata di chiunque possa accedervi, in qualsiasi circostanza.
I proprietari terrieri possono creare la propria scena tramite lo strumento Decentraland Builder e programmare diverse attività tramite l'SDK Decentraland. Diversi gruppi si sono organizzati come "società", che hanno messo in comune appezzamenti di terreno nella speranza di creare distretti a tema. Alcuni degli sviluppi pianificati attraverso le proprietà esistenti includono quartieri dello shopping, parchi di divertimento, università e quartieri del gioco d'azzardo.
I proprietari terrieri esistenti possono scambiare i loro lotti con MANA, con una percentuale di ogni transazione bruciata automaticamente. Dei circa 2,6 miliardi di fornitura iniziale di MANA, oltre 1,5 miliardi di token sono già stati bruciati o altrimenti congelati. Questa meccanica di combustione aggressiva ha indotto gli speculatori ad accumulare MANA e appezzamenti di terreno in previsione di ulteriore scarsità. Gli appezzamenti di terreno più economici sono attualmente valutati intorno a 10.000 MANA, o circa 400 USD. Gli appezzamenti più attraenti, come le parcelle centrali o lungo la strada, e i raggruppamenti sono comunemente venduti a una tariffa dieci volte superiore e talvolta anche di più. È difficile stabilire un prezzo per le proprietà organizzate, sebbene questi distretti più grandi valgano potenzialmente centinaia di migliaia o addirittura milioni di dollari. L'attuale capitalizzazione di mercato di MANA è di circa 1 miliardo di euro. Appassionati e speculatori hanno posto molta enfasi su Decentraland per il suo approccio unico al gioco blockchain. Decentraland si distingue dai giochi esistenti e falliti su Ethereum in quanto aspira a creare qualcosa che si basi sulla blockchain per avere successo. Un universo di realtà virtuale autonomo, interamente posseduto e controllato dai giocatori è un'impresa enorme attualmente impossibile entro i limiti dell'industria del gioco tradizionale.
Decentraland sta anche ricevendo supporto da altri giocatori nello spazio di gioco di Ethereum. My Crypto Heroes distribuisce equipaggiamento da collezione che gli avatar possono indossare. Altre DApp basate su Ethereum hanno acquistato i propri appezzamenti di terreno nella speranza di instaurare una interoperabilità.

Figura 11: Visita di un museo su Decentraland

6.3 The Sandbox

Un videogioco che ultimamente ha ricevuto particolare notorietà è The Sandbox. Esso offre la possibilità di creare, gestire e giocare esperienze ed avventure tramite l'utilizzo di una blockchain, dove viene registrata ogni operazione eseguita sulla piattaforma di gioco. Il primo prodotto offerto dal videogioco, presentato dopo un periodo di sviluppo durato 3 anni, è una mappa formata da più di 160 mila appezzamenti di terra digitali.

Figura 12: La mappa di Sandbox aggiornata a Giugno 2021

Possiamo paragonare questi lotti a terreni reali, su cui però si possono compiere varie azioni, come ad esempio costruire un edificio caratterizzato da scenari o servizi diversi in base all'esperienza di gioco che si vuole offrire.
Per capire il motivo per cui alcuni appezzamenti possano valere più o meno di altri, si può pensare, ad esempio, ad un gioco come Minecraft, dove i giocatori passano ore a reperire materiali esplorando la mappa, per poi costruire edifici di varia complessità e bellezza. I creatori di The Sandbox, dunque, hanno voluto dare la possibilità ai giocatori di poter vendere le proprie creazioni, dando ad esse una concezione di opere d'arte vere e proprie, di pezzi unici dal valore più o meno alto. Ecco che quindi i lotti diventano a tutti gli effetti degli NFT che tanto vanno di moda negli ultimi anni.
Molte società di varia natura hanno già iniziato ad investire sull'acquisto di terreni virtuali su The Sandbox, come ad esempio Atari (produttore di videogiochi e console), ma anche Coinmarketcap e Binance (esperti di criptovalute).

Il prezzo di questi lotti varia in base a diversi fattori, ad esempio alcuni punti della mappa sono più "strategici" di altri e assumono più valore; un altro fattore può essere il fatto di possedere un lotto vicino a un'azienda importante, perché in questo modo lo si può offrire ad eventuali acquirenti vantandosi della vicinanza a prodotti e servizi, come quando, nel mondo reale, si convince più facilmente una persona ad acquistare un appartamento in centro città piuttosto che in periferia. In entrambi i casi, la vicinanza ai servizi (che siano essi fisici o digitali) fa la differenza sia per quanto riguarda il prezzo, che per quanto riguarda l'appetibilità.
Le potenzialità di questo gioco sono enormi, si pensi solo che il suo "parente" Minecraft conta circa 150 milioni di giocatori. Se nel prossimo futuro The Sandbox venisse "abitato" da una popolazione di uguale grandezza o maggiore grazie alla blockchain, sarebbe possibile scambiare gli oggetti, i terreni o qualsiasi bene virtuale in gioco come se fossero asset e potrebbe emergere un'economia più grande dell'intero Giappone, come afferma il fondatore della società di social gaming Gumi, Hironao Kunimitsu. Secondo il suo punto di vista, cambierebbe anche il modo in cui la società si approccia ai videogiochi: i genitori attualmente sgridano i propri figli perché passano il tempo davanti ai videogiochi, ma se le ore passate a giocare servissero a creare pezzi unici da poter scambiare per del valore economico, forse verrebbero addirittura incoraggiati a farlo.

7 Trend del settore del Blockchain gaming

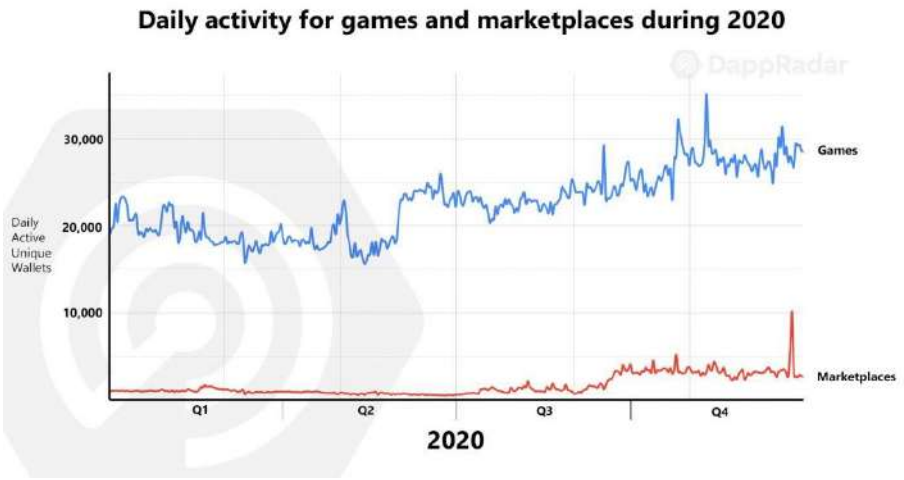
Il 2020 è stato un anno significativo per molte categorie nel settore delle DApp su blockchain, ma quando parliamo di gaming su blockchain la situazione diventa più sfumata.

7.1 Punti chiave

- Le attività giornaliere sui giochi blockchain è cresciuta del 35% durante il 2020 fino a 28,000 DAUW (Daily Active Unique Wallet).
- Il volume di trading sui giochi blockchain durante il 2020 è aumentato del 191%, totalizzando 54 milioni di dollari.
- L'attività giornaliera nei marketplace di NFT è cresciuta del 226% fino a 3,400 DAUW.
- Il volume di trading nei marketplace di NFT è aumentato del 785%, totalizzando 78 milioni di dollari.
- Ethereum è stata la principale fonte del volume di trading: 61% dei giochi e 92% dei marketplace.

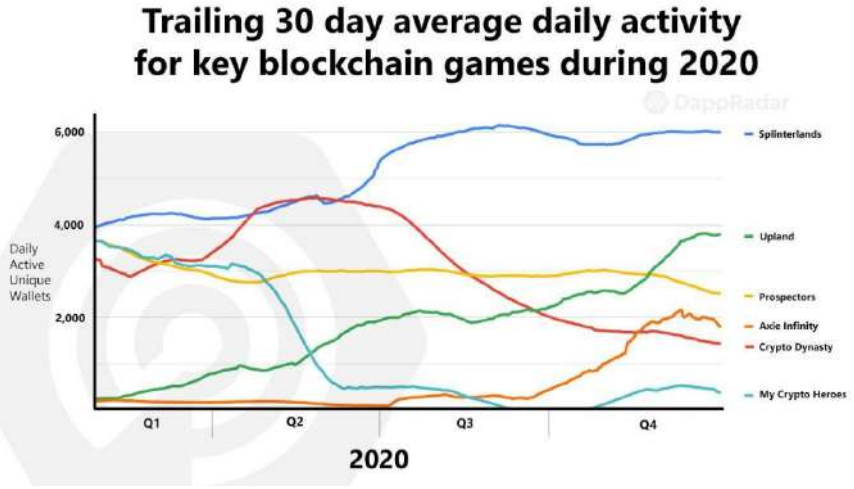
7.2 Un anno di crescita

Sicuramente c'è stata una crescita, ma comparata con quella della categoria DeFi, che è stata il catalizzatore per l'intero settore delle criptovalute (15 miliardi di dollari di locked value alla fine dell'anno), la crescita è stata irregolare e sostenuta soprattutto dalle dinamiche dei singoli prodotti piuttosto che da tutto il settore.



Il totale dell'attività media giornaliera dei wallet su tutti i giochi blockchain monitorata da DappRadar durante dicembre è stata il 35% in più rispetto a gennaio 2020.
Ciò è avvenuto nonostante il generale declino delle attività per i giochi su Ethereum da maggio in avanti, quando le tasse di transazione (gas fee) sono diventate troppo costose a causa del boom del settore DeFi. Da gennaio a dicembre le attività giornaliere su tutti i giochi basati su Ethereum è crollata del 45%.
DappRadar al momento monitora più di 700 giochi basati su 17 diverse blockchain che utilizzando smart contract.
Se si analizza gioco per gioco però, la situazione risulta radicalmente diversa.

- Axie Infinity (Ethereum) è cresciuto dell' 810%
- Uplands (EOS) è cresciuto del 516%
- Splinterlands (Hive) è cresciuto del 53%
- Prospectors (EOS & WAX) è diminuito del 31%
- Crypto Dynasty (EOS) è diminuito del 48%
- My Crypto Heros (Ethereum) è diminuito del 97%



In vari momenti durante il 2020, tutti questi titoli sono stati tra i primi giochi più popolari, il che dimostra la volatilità della partecipazione del pubblico a questi giochi. Durante il 2020, EOS è stata la blockchain più popolare per il gaming, avendo il 35% di tutti i wallet attivi, rispetto al 17% su Hive, 14% su Ethereum, 10% su Thundercore e 9% su Wax.

7.3 Conclusioni

In conclusione si può dire che, coerentemente con il mondo delle criptovalute, anche la partecipazione ai giochi su blockchain è molto volatile. Ci sono infatti ancora molti fattori che hanno un impatto considerevole su questo settore. Il primo fattore, come visto in precedenza, riguarda il fatto che il settore dei giochi blockchain è stato trainato fortemente dal grandissimo successo delle applicazioni DeFi (soprattutto dall'inizio del 2020). Un secondo fattore riguarda il fatto che il settore dei giochi blockchain è ancora molto legato alla blockchain di Ethereum: un aumento delle gas fee può diminuire drasticamente la partecipazione a questi giochi.
La volatilità della partecipazione a questo settore è probabilmente causato dalla sua giovinezza, ma le potenzialità che ha sono enormi, e in futuro ci si aspetta uno scenario caratterizzato da una partecipazione maggiore e sempre più stabile.

Riferimenti bibliografici

[1] 3.151 euro al minuto per gioco: l'industria dei videogiochi non conosce crisi - proiezioni di borsa.

[2] Blockchain gaming in 2020.

[3] Blockchain gaming part i: The opportunity.

[4] Blockchain gaming part ii: Successes and failures of first-generation games.

[5] Blockchain gaming part iii: Protocol-led second-generation games.

[6] Blockchain gaming part iv: The road ahead.

[7] Casi d'uso della blockchain: Gaming — binance academy.

[8] Cos'è decentraland (mana): il mondo virtuale su blockchain.

[9] Decentraland: ecco il mondo virtuale che fa monetizzare.

[10] How blockchain is making digital gaming better - blockchain pulse: Ibm blockchain blog.

[11] Il videogioco su blockchain darà vita a un'economia più grande dell'intero giappone.

[12] Industria videogiochi 2020, raggiunti $ 159 miliardi di fatturato.

[13] The sandbox, il gioco di compravendita di terreni digitali sulla blockchain - la stampa.

[14] Top gaming tokens by market capitalization — coinmarketcap.

[15] Utilizzare il blockchain nel settore del gaming - investire come i migliori.

[16] Tian Min, Hanyi Wang, Yaoze Guo, and Wei Cai. *Blockchain Games: A Survey.*

CARDANO

Giulia Pititto, Daniele Plutino, Alessandro Rigoli, Luca Saglia

Indice

Capitolo 1

Introduzione

Cardano è una piattaforma blockchain decentralizzata di terza generazione che nasce, nel 2015, dall'incontro tra filosofia scientifica e ricerca; proprio per questo prende il nome dal matematico italiano Girolamo Cardano.

La piattaforma Cardano è stata progettata da zero sotto la guida di numerose figure nell'area dell'ingegneria e da esperti di blockchain e crittografia.
Sono tre le società che supervisionano lo sviluppo del progetto e del suo ecosistema:

- Input Output Hong Kong (IOHK), il cui Chief Executive Officer (CEO) è Charles Hoskinson, ex co-fondatore di Ethereum, a capo del progetto Cardano;
- Emurgo, responsabile della promozione dell'adozione di Cardano attraverso iniziative commerciali;
- Fondazione Cardano, responsabile del monitoraggio dello sviluppo del progetto e della modellizzazione degli standard legislativi e commerciali.

Cardano presta molta attenzione alla sostenibilità, alla scalabilità e alla trasparenza. È un progetto completamente open source che mira a fornire un'infrastruttura inclusiva, equa e resiliente per applicazioni finanziarie e sociali su scala globale. Uno dei suoi obiettivi principali è offrire servizi finanziari affidabili e sicuri a coloro che attualmente non ne hanno accesso.

La criptovaluta utilizzata da Cardano è ADA, che prende il nome da Augusta Ada Byron, meglio nota come Ada Lovelace, considerata la prima programmatrice di computer al mondo. Qualsiasi utente, situato in un luogo qualsiasi del mondo, può utilizzare ADA come valore di scambio sicuro, senza la necessità di terzi che medino lo scambio. Ogni transazione è registrata in modo permanente, sicuro e trasparente sulla blockchain di Cardano. Ogni titolare di ADA detiene anche una partecipazione nella rete Cardano: gli ADA immagazzinati nel portafoglio possono essere delegati a un pool di stake per guadagnare ricompense,

ottenute per aver preso parte alla gestione della rete.
Con il tempo ADA sarà anche utilizzabile per varie applicazioni e servizi sulla piattaforma Cardano.

Al momento, il prezzo di mercato di Cardano è $1.57952994, con una capitalizzazione di mercato di $50.46 miliardi. L'offerta totale disponibile di Cardano è 31.95 miliardi di ADA e si trova al quinto posto nel mercato delle criptovalute. Il prezzo di ADA è salito del 7.33% nelle ultime 24 ore (al 25 Maggio ore 19:50).

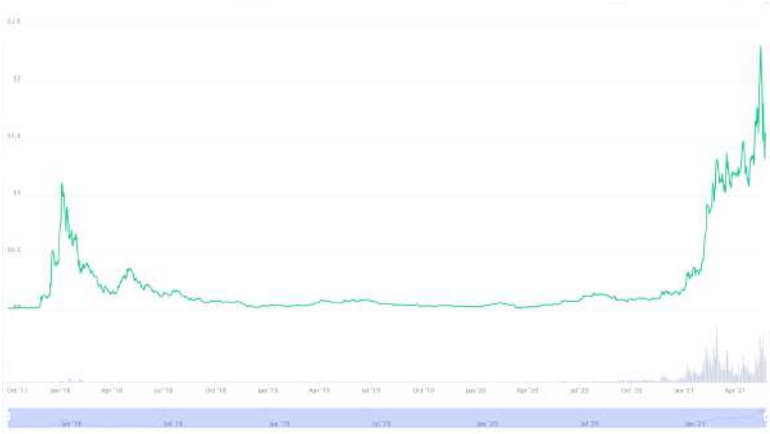


Figura 1.1: Andamento prezzo moneta ADA da Ottobre 2017 a Maggio 2021

La catena blockchain di Cardano è divisa in *epoche*, a loro volta suddivise in un insieme di *time slots*. Si stima che un'epoca venga divisa in 432 mila time slots. Ogni time slot dura 1 secondo, quindi, secondo il rapporto di prima, un'epoca dura 5 giorni. Ogni blocco ha diverse informazioni al proprio interno, come l'ID del blocco, l'epoca e il time slot a cui appartiene, il numero di transazioni nel blocco, ecc., e si possono vedere anche i dettagli di ogni transazione.

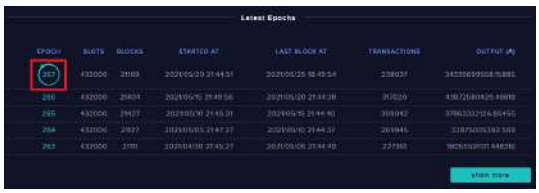


Figura 1.2: Catena blockchain

Figura 1.3: Blocco 5765280

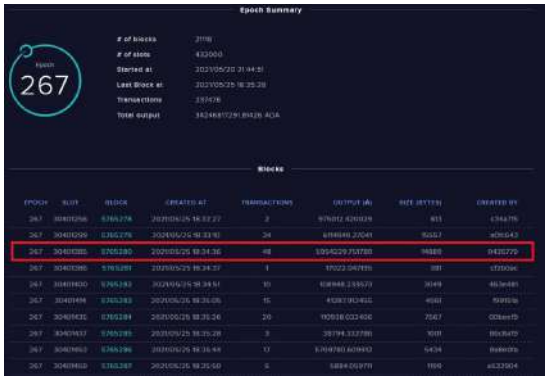


Figura 1.4: Epoca 267

Figura 1.5: Transazione

Il percorso di sviluppo di Cardano è stato suddiviso in cinque ere, ciascuna incentrata su un tema principale:

- Byron - fondazione
- Shelley - decentralizzazione
- Goguen - smart contracts
- Basho - scalabilità
- Voltaire - governance

Ciascuna era è incentrata su un insieme di funzionalità che vengono sviluppate in più versioni di codice. Sebbene questi flussi di sviluppo vengano forniti in sequenza, il lavoro per ciascuno avviene in parallelo, con ricerca, prototipazione e sviluppo spesso in contemporanea.

Figura 1.6: Le 5 ere di Cardano

Capitolo 2

Era Byron

L'inizio dell'era Byron è stata sancita dal rilascio, a Settembre 2017, della prima versione di Cardano. Questa era prende il nome dal poeta britannico George Gordon Noel Byron, detto Lord Byron, nonché padre di Ada Lovelace. Tale era aveva tra gli scopi principali:

- la creazione della criptomoneta ADA;
- la definizione della prima versione del protocollo di consenso Ouroboros;
- il rilascio dei due *wallet* ufficiali;
- la creazione della Cardano testnet;
- la nascita di una comunità di persone coinvolte nel processo di sviluppo del progetto.

2.1 *Wallet*

Con l'inizio dell'era Byron, Cardano ha anche rilasciato le prime versioni dei suoi *wallet* ufficiali: Daedalus, che è il wallet desktop ufficiale di IOHK per tenere ADA, e Yoroi, un *light wallet* disegnato da Emurgo per le transazioni veloci e l'uso quotidiano.

2.1.1 *Daedalus*

Daedalus, sviluppato da IOHK, è un *wallet* che permette di depositare ADA; il sito stesso di IOHK definisce Daedalus come un "*desktop, full-node hierarchical deterministic (HD) wallet*".
Essendo un *wallet* HD, Daedalus genera automaticamente le coppie di chiavi e gli indirizzi in una struttura gerarchica ad albero. Ogni account Daedalus corrisponde ad un *full-node* della rete di Cardano e, pertanto, Daedalus conserva l'intera blockchain di Cardano e valida in modo indipendente tutti i blocchi e

le transazioni della catena. Proprio per questo il download ed il setup iniziale di Daedalus richiedono da una a due ore e l'installazione necessita di diversi giga di memoria disponibili (almeno 6.5 GB). Un singolo account di Daedalus permette di gestire fino a 20 diversi *wallet*, le cui chiavi private sono salvate solo in locale e non condivise con terze parti. L'utente deve soltanto ricordare la frase mnemonica da 12 o 24 parole, a seconda della versione scaricata.

2.1.2 *Yoroi*

Yoroi, al contrario di Daedalus, è un *wallet* leggero (*light*) per la criptomoneta ADA e può essere scaricato come estensione di Chrome o di Cardano. Un account browser Yoroi permette di gestire un unico *wallet* il quale viene connesso tramite Emurgo a un *full-node* della rete di Cardano, di cui "si fida"; è possibile gestire più *wallet* solo dall'app mobile. Al contrario di Daedalus, il download ed il setup iniziale di Yoroi sono istantanei, in quanto esso non conserva tutta la catena in memoria. Pertanto, anche lo spazio in memoria necessario per gestire un *walllet* Yoroi è minimo (circa 6.5 Mb). Questo rende Yoroi particolarmente adatto agli utenti che hanno intenzione di fare molte transazioni in poco tempo. Anche per Yoroi la chiave privata è salvata solo in locale e non condivisa con terze parti; in questo caso la frase mnemonica da ricordare può essere di 15 o 24 parole.

2.2 *Firma digitale*

Come qualsiasi criptomoneta anche Cardano ha bisogno di utilizzare almeno uno schema di firma digitale per garantire la sicurezza delle transazioni sulla blockchain. Per stabilire quali fossero gli schemi migliori, gli sviluppatori hanno tenuto conto di due aspetti condizionanti: la durata della sicurezza dello schema nel lungo periodo, ovvero quanto tempo si prevede che passi prima che lo schema venga rotto; la possibilità che imprese, governi e istituzioni abbiano degli schemi preferiti o, addirittura, obbligatori. Nel caso in cui una criptomoneta scelga di usare un solo schema di firma digitale, essa accetta che quello possa essere rotto in futuro o che alcuni enti possano non usarlo. D'altronde, il numero di schemi di firma deve rimanere contenuto, poiché ogni client deve essere in grado di capire e validare ciascuno schema. Sulla base di queste riflessioni, nel proprio whitepaper, i fondatori di Cardano hanno auspicato l'utilizzo dei seguenti schemi di firma digitale all'interno della blockchain:

- EdDSA con hash SHA-512 sulla curva ellittica di Edwards 25519 (anche indicato con Ed25519) come schema di firma di partenza;
- BLISS-B con lo scopo di introdurre uno schema di firma in grado di resistere ai quantum computer;
- ECDSA sulla curva ellittica secp256k1 con l'obiettivo di favorire l'interoperabilità con altre criptomonete come Bitcoin.

Il primo dei tre schemi di firma digitale sopra elencati è l'unico attualmente in uso sulla blockchain di Cardano. Analizziamo in dettaglio come funziona dal punto di vista crittografico.

2.2.1 *EdDSA*

Per spiegare come funziona l'algoritmo di firma Ed25519 è necessario introdurre alcuni concetti di geometria algebrica.

Definizione 2.2.1 (Curva di Edwards)**.** *Sia dato un campo K tale che $char(K) \neq 2$; si dice curva di Edwards su K una curva ellittica della forma*

$$x^2 + y^2 = 1 + dx^2y^2,$$

dove $d \in K \setminus \{0, 1\}$.

Definizione 2.2.2 (Curva di Edwards twisted)**.** *Sia dato un campo K tale che $char(K) \neq 2$; si dice curva di Edwards twisted su K una curva ellittica piana affine della forma*

$$ax^2 + y^2 = 1 + dx^2y^2,$$

dove $a, d \in K \setminus \{0\}$ e $a \neq d$.

EdDSA (Edwards-curve Digital Signature Algorithm) è uno schema asimmetrico di firma digitale basato sulla difficoltà del problema del logaritmo discreto su curve ellittiche, il quale utilizza una variante della firma di Schnorr sulle curve di Edwards twisted. In particolare, per costruire uno schema EdDSA sono necessari:

- un campo finito $\mathbb{F}_q$ su un primo q;
- una curva di Edwards twisted E su $\mathbb{F}_q$, il cui gruppo dei punti $\mathbb{F}_q$-razionali abbia ordine $|E(\mathbb{F}_q)| = 2^c l$, dove l è un primo grande e 2^c è detto cofattore;
- un punto generatore $G \in E(\mathbb{F}_q)$ di ordine l;
- un intero b tale che $2^{b-1} > q$ (consigliato multiplo di 8), in modo che i punti di $E(\mathbb{F}_q)$ possano essere rappresentati da stringhe di b bits;
- una funzione hash H con output di $2b$ bits.

Generazione delle chiavi

Dati questi elementi, una coppia di chiavi dello schema è costituita, così come in ECDSA, da:

- una chiave privata k costituita da un intero di b bits;
- una chiave pubblica P costituita da un punto di $E(\mathbb{F}_q)$.

La chiave privata viene generata da un intero casuale di lunghezza opportuna, detto *seed*, tramite un algoritmo deterministico che coinvolge l'hash del *seed*. La chiave pubblica è calcolata come $P = k * G$, ma spesso viene utilizzata in forma compressa concatenando il bit di parità della coordinata x alla coordinata y.

Algoritmo di firma

L'algoritmo di firma prende in input un messaggio M e la coppia di chiavi (k, P) di colui che firma e restituisce in output la firma costituita dalla coppia (R, s). L'algoritmo funziona nel modo seguente:

- si genera deterministicamente un intero segreto $r = H(H(k) + M) \bmod q$;
- si calcola $R = r * G$;
- si calcola $h = H(R + P + M) \bmod q$;
- si calcola $s = (r + h * k) \bmod q$;
- viene restituita la firma data dalla coppia (R, s).

Verifica della firma

L'algoritmo di verifica della firma digitale prende in input il messaggio M, la chiave pubblica del firmatario P e la firma (R, s). L'algoritmo si articola nel modo seguente:

- si calcola $h = H(R + P + M) \bmod q$;
- si calcola $U1 = s * G$;
- si calcola $U2 = R + h * P$;
- si verifica se $U1 = U2$.

La firma funzione perché

$$U1 = s * G = [(r + h * k) \bmod q] * G = r * G + h * k * G = R + h * P = U2$$

Ed25519

La blockchain di Cardano utilizza la firma Ed25519 che consiste nell'algoritmo EdDSA sulla curva ellittica 25519. Dunque i parametri scelti da Cardano per implementare l'algoritmo sono:

- $q = 2^{255} - 19$, da cui il nome della curva;
- $E(\mathbb{F}_q) : -x^2 + y^2 = 1 - \frac{121665}{121666} x^2 y^2$;
- $l = 2^{252} + 27742317777372353535851937790883648493$ e $c = 3$;
- G è l'unico punto di $E(\mathbb{F}_q)$ con coordinata $y = \frac{4}{5}$ e x positiva;
- $b = 256$;
- funzione hash H data da SHA-512.

2.3 *Da Byron a Shelley: Hard Fork Combinator*

La transizione dall'era Byron, che ha gettato le basi di Cardano, alla sua successiva, ovvero l'era Shelley, si è realizzata attraverso una vera e propria innovazione introdotta da Cardano, quella dell'hard fork combinator.

Il termine *hard fork* è generalmente usato per descrivere un cambiamento radicale all'interno di una blockchain, che si traduce in un cambiamento di uno o più aspetti chiave della rete, come ad esempio l'implementazione del protocollo di consenso, il contenuto dei blocchi o della loro creazione. Questo porta ad una totale scissione tra la blockchain *pre-fork* e la blockchain *post-fork* (troviamo molti esempi di questo in letteratura, come nel caso di Bitcoin con Bitcoin cash, classic, ecc...), con conseguente scissione dell'utenza: i nodi aggiornati non avranno retro compatibilità con quelli non aggiornati.

A differenza della precedente, un *hard fork combinator* consente di effettuare un aggiornamento di entità pari a quello effettuato attraverso un hard fork, ma non presentando la scissione tra blockchain pre e post hard fork: i protocolli, anche se diversi tra loro, possono coesistere e la blockchain non presenta alcun tipo di interruzione.

2.3.1 *Hard Fork Combinator - funzionamento*

L'hard fork, nella sua implementazione tradizionale, prevede che, a partire da un certo blocco, ci sia un cambiamento radicale: se, per esempio, avviene nel protocollo, da quel momento si utilizzerà solo la nuova versione, a danno di tutti i nodi che non hanno completato, volontariamente o per mancanza di tempo, l'aggiornamento.

L'hard fork combinator introduce invece la possibilità di mantenere, per un periodo di transizione, sia la nuova versione del protocollo che quella precedente. I nodi possono condividere informazioni su quale sia l'ultima versione supportata e, inoltre, il meccanismo di accordo sulla versione di protocollo da utilizzare è computazionalmente sostenibile.

L'upgrade di un nodo non avviene in modo automatico, ma si basa su un meccanismo di proposta e approvazione: deve essere inviata una richiesta alla rete, che può essere approvata o meno e, se questa raggiunge una certa soglia di approvazione, il nodo può procedere con l'aggiornamento. Il cambiamento sarà effettivo a partire dall'epoca successiva a quella della proposta. Si osserva che, proposte inviate a ridosso della fine di un'epoca vengono automaticamente scartate, per motivi di sicurezza. Quando la totalità o un numero sufficiente di nodi ha effettuato l'upgrade, la transizione si rende effettiva nell'intera rete, diventando lo standard.

È importante sottolineare come il vantaggio principale dell'hard fork combinator risieda nella coesistenza di diversi upgrade di un protocollo: il mancato funzionamento della nuova versione non comprometterebbe il funzionamento dell'intera rete, in quanto si potrebbe comunque continuare a comunicare attraverso la vecchia, fino a completa stabilizzazione.

2.3.2 *Primi tentativi di utilizzo*

Nel caso di Cardano si evidenzia un solo evento di hard fork tradizionale che ha portato dall'implementazione del protocollo di consenso Ouroboros Classic a Ouroboros BFT: in questo caso è stato necessario far ripartire la mainnet con la nuova implementazione del protocollo.
Il primo evento di hard fork combinator si è svolto nel passaggio tra le ere Byron e Shelley, con ottimi risultati e in maniera graduale: la sostituzione di Ouroboros BFT con Ouroboros Praos non ha richiesto l'aggiornamento in contemporanea di tutti i nodi, ma per un periodo limitato di tempo i due protocolli sono coesistiti, così come nodi aggiornati e non, il tutto nel pieno funzionamento della rete.

In futuro, l'hard fork combinator verrà utilizzato anche per le transizioni alle successive ere.

Capitolo 3

Era Shelley

L'era Shelley rappresenta l'inizio del processo di decentralizzazione da parte di Cardano: la rete federata dell'era Byron, dove i nodi erano esclusivamente gestiti da IOHK, viene gradualmente disattivata per far posto a sempre più nodi di proprietà degli utenti della piattaforma, in un'ottica di maggiore decentralizzazione, sicurezza e robustezza. Inoltre, viene introdotto il meccanismo di delega e degli incentivi, per una più vasta adozione da parte degli utenti.
Per la fine dell'era Shelley, prevista a breve, ci si aspetta una decentralizzazione 50-100 volte maggiore rispetto alle altre blockchain più importanti, come Bitcoin o Ethereum: circa 1000 nuovi nodi di proprietà degli utenti contro le decine di grandi mining pool delle altre piattaforme, che introducono un maggior rischio di comportamenti scorretti.
L'innovazione principale di quest'era è rappresentata dall'introduzione dell'aggiornamento del protocollo di consenso Ouroboros in Ouroboros Praos, con il supporto all'inserimento dei blocchi non più soltanto dai nodi federati, ma anche e soprattutto dai nodi di proprietà degli utenti.

3.1 *Ouroboros*

Come tutte le blockchain, anche Cardano ha bisogno di un protocollo di consenso. La scelta, in questo caso, non si è rivolta verso una più classica e dispendiosa, in termini di energia, Proof of Work (PoW), come in Bitcoin ed Ethereum, ma piuttosto verso una soluzione di tipo Proof of Stake (PoS). L'implementazione attuale della PoS specifica per Cardano è Ouroboros Praos; tuttavia questo è solo l'ultimo di una serie di aggiornamenti del protocollo, che sono:

- Ouroboros Classic
- Ouroboros BFT
- Ouroboros Genesis
- Ouroboros Praos

Segno del continuo aggiornamento effettuato su questa piattaforma, è già stata presentata una nuova implementazione del protocollo che in futuro rimpiazzerà Praos, chiamata *Ouroboros Hydra*.

3.1.1 *Concetti base*

Ouroboros, nell'implementazione attuale Praos, basa il suo funzionamento su una suddivisione temporale in epoche e time slots. Per ogni time slot, zero o più nodi produttori di blocchi vengono eletti come *slot leaders* in maniera pseudo-casuale; in media ne viene eletto uno ogni 20 secondi. A questo punto gli slot leaders hanno il compito di creare un blocco: tra questi se ne seleziona uno randomicamente mentre gli altri vengono scartati. Lo slot leader ha anche il compito di validare le transazioni, verificando che chi invia denaro abbia sufficienti fondi e che i parametri della transazione vengano rispettati. In caso positivo li aggiunge al blocco da inserire.

3.1.2 *Stake pools e Stakeholders*

Ouroboros distingue due figure chiave con un ruolo attivo per il funzionamento del protocollo, che sono gli *stakeholders* e gli *stake pools*:

- appartengono alla categoria di stakeholders tutti coloro che possiedono un "interesse" (stake) all'interno della blockchain, misurato in maniera proporzionale alla quantità di ADA posseduti.
  Uno stakeholder U_i possiede una coppia di chiavi (vk_i, sk_i) chiamate rispettivamente "chiave di verifica" e "chiave di firma": la prima necessaria per l'identificazione univoca dello stakeholder all'interno della rete, la seconda per la firma di eventuali blocchi inseriti. Viene memorizzato, inoltre, il numero di stakes S_i posseduti e anche informazioni ausiliarie ρ, che vengono utilizzate per il processo di elezione dello slot leader;

- gli stake pools sono dei nodi all'interno della rete di Cardano che assumono un duplice compito: l'inserimento di nuovi blocchi, qualora vengano scelti come slot leader, e la raccolta delle deleghe di stake da parte degli stakeholders.

Di fatto, soltanto uno stake pool può essere eletto come slot leader, ma ogni stakeholder potenzialmente può creare il proprio stake pool: qualsiasi utente della rete, sotto opportune condizioni, può inserire un blocco nella rete.

3.1.3 *Delega e selezione di uno stake pool*

Uno stakeholder che vuole partecipare attivamente al protocollo per guadagnare ADA come ricompensa per l'inserimento di un blocco ha due modi per farlo: il più comune è delegare il proprio stake a uno stake pool esistente, l'altro è di creare il proprio stake pool e raccogliere il maggior numero di stake possibili dagli altri utenti, in quanto maggiore è il numero di stake delegati, maggiore è

la probabilità di essere selezionati come slot leader.
La scelta di un pool come slot leader avviene in maniera pseudo-casuale, pseudo in quanto la probabilità di essere selezionati aumenta con l'aumentare del numero di stake delegati al pool.

Delega degli stake

Il processo di delega è quello per cui uno stakeholder "affida" una porzione o tutti i propri stake (associati agli ADA posseduti) ad uno stake pool. Qualora questo venisse eletto come slot leader e riuscisse ad inserire il nuovo blocco ottenendo una ricompensa, questa verrebbe divisa tra tutti gli stakeholders partecipanti al pool, in maniera proporzionale alla quantità di stake delegati. È bene evidenziare alcuni aspetti legati a questo processo:

- in caso di mancato inserimento del blocco o di mancata elezione del pool come slot leader, il delegante non perde gli stake/ADA posseduti, semplicemente non otterrà alcuna ricompensa;
- utilizzando i wallet ufficiali di Cardano, è possibile delegare porzioni di stake a pool diversi dallo stesso wallet soltanto attraverso Yoroi, mentre con Daedalus è necessario creare wallet differenti. Per ogni wallet creato si può selezionare un pool e parteciparvi in misura proporzionale agli ADA contenuti in quel wallet. È importante sottolineare che in futuro la delega multi-pool da un singolo wallet non verrà più supportata;
- è possibile ri-delegare stake già assegnati ad un pool in ogni momento, tuttavia la scelta del pool sarà valida a decorrere dall'epoca successiva a quella del cambio di preferenza;
- gli ADA eventualmente guadagnati dalla delega verranno aggiunti al wallet da cui si è delegato e incrementeranno anche l'ammontare di stake posseduti;
- per una più semplice e migliore scelta da parte degli stakeholders, gli stake pool vengono inseriti in una classifica basata sulla "bontà" del pool.

Realizzare uno stake pool

Come detto precedentemente, la seconda possibilità che si presenta ad un utente della rete è quella di creare il proprio stake pool, ma questo può avvenire soltanto se vengono rispettati alcuni vincoli. In particolare, ad uno stake pool operator sono richieste:

- conoscenze su come eseguire e mantenere un nodo Cardano 24 ore su 24, 7 giorni su 7;
- conoscenze approfondite sul funzionamento del protocollo;
- esperienza in ambito DevOps (development and operations);

- conoscenze sulla gestione dei server.

Mentre per quanto riguarda l'infrastruttura, è necessario avere:

- una connessione internet stabile e affidabile;
- dispositivi con un sistema operativo compatibile installato: linux (2.6.18 o succ.), BSD (NetBSD 8.x o FreeBSD 12.x), OSX (10.7 o succ.), Windows 10.

Anche in questo caso, è importante sottolineare alcuni aspetti legati al meccanismo di creazione di un pool:

- in fase di creazione del pool, il creatore deve stabilire la sua percentuale di guadagno sulla ricompensa ottenuta dalla creazione di un blocco e la percentuale che, invece, deve essere ripartita tra i deleganti. Ovviamente il creatore deve tenere conto che questo è determinante nell'"appetibilità" di un pool.
- Tecnicamente, è possibile realizzare uno stake pool privato impostando, in fase di creazione, la percentuale di guadagno del creatore al 100%, così da scoraggiare gli altri utenti a delegare proprio a questo pool, ma lasciando la libertà al creatore di delegare i propri stake al suo interno. Generalmente viene fatto in fase di testing o di update.
- Anche se si esegue il proprio stake pool, la piattaforma non garantisce alcun guadagno: si otterrà una ricompensa esclusivamente attraverso l'inserimento di un blocco.
- Il creatore dello stake pool ha una responsabilità nei confronti dei suoi deleganti, nonché della salute dell'intera rete.

3.1.4 *Stake pool: parametri chiave*

Come si può intuire dai punti sopra, per uno stake pool è fondamentale raggiungere buoni risultati nei parametri di valutazione, per scalare la classifica ed essere più appetibile agli stakeholders, nonché per un maggiore guadagno per l'inserimento di un blocco. Approfondiamo di seguito questi parametri.

A0 - Meccanismo di impegno (pledging)

Il parametro A0 misura l'"impegno" dei pool operators all'interno del pool. È un concetto simile all'auto-staking, ovvero un operator che delega i propri stake al proprio pool; tuttavia, questo viene effettuato prima ancora che sia possibile lo staking, durante la fase di creazione, a testimonianza dell'impegno finanziario di chi ha creato il pool. Il concetto che sta alla base di questo parametro è che un pool con più stake "impegnati" è più attraente per uno stakeholder per due motivi:

1. il possessore del pool è finanziariamente motivato a mantenere questo attivo e non perdere nessuno slot;

2. gli ADA impegnati con questo meccanismo sono un parametro per decidere la retribuzione di un pool per l'inserimento di un blocco.

Indice di desiderabilità

Questo indice quantifica l'appetibilità di un pool per un ADA holder. È un parametro influenzato da numerosi fattori, tra cui:

- performance: misura le prestazioni attuali di un pool in base a quanti blocchi ha attualmente prodotto in proporzione a quante volte è stato eletto come slot leader. Ad esempio, se un pool inserisce la metà dei blocchi che avrebbe potuto inserire, il suo punteggio in termini di performance è pari al 50%; questo può succedere perché il pool ha una scarsa connessione internet o perché l'operator ha messo il pool offline per un certo periodo di tempo.

- Margine di guadagno dell'owner: quanta percentuale dei guadagni dati dall'inserimento di un blocco viene trattenuta dal possessore del pool, parametro impostato in fase di creazione.

- Guadagni totali derivanti dall'inserimento del blocco attuale.

- Guadagno totale ottenuto nell'epoca attuale.

Questo indice viene utilizzato per effettuare il ranking dei pool.

K - parametro di saturazione

Il guadagno ottenuto da un pool per l'inserimento di un blocco è proporzionale al numero di stake ad esso delegati. Questo, però, potrebbe portare a un problema di centralizzazione: tanti più stake vengono delegati a un pool, tanto più guadagno esso avrà, così come sarà maggiore la probabilità di essere scelto come slot leader e, di conseguenza, la desiderabilità per uno stake holder. Si creerebbe, quindi, un circolo che porterebbe tutti alla scelta del medesimo pool. Il parametro K serve proprio ad evitare questo scenario: quando il numero di stake delegati ad un pool raggiunge il punto di saturazione, inizierà ad offrire una percentuale di guadagno sempre minore agli stake holders che continueranno a delegare ad esso, per incentivare la scelta di pool minori e avere così una maggiore decentralizzazione.
Questo parametro serve per salvaguardare gli interessi sia dei pool che degli holder.

d - parametro di decentralizzazione

Per capire a fondo questo parametro, occorre riportare una differenza sostanziale tra la prima e la seconda fase di Cardano: durante la prima era (Byron), nell'ottica di una blockchain nascente, i blocchi venivano inseriti in maniera centralizzata dai nodi IOHK ($d = 1$). Tuttavia Cardano, come la maggior parte delle blockchain, mira alla decentralizzazione, per cui auspica alla gestione del protocollo esclusivamente da parte degli utenti, ovvero soltanto dai pool ($d = 0$). Il parametro di decentralizzazione inserito e settato a 1 durante l'evento di hard fork combinator, che ha portato dall'era Byron a quella Shelley, è un parametro regolabile che specifica la percentuale di blocchi assegnati a nodi federati rispetto a quanti vengono assegnati a pool creati da utenti. La sua creazione è stata necessaria per consentire alla rete di stabilizzarsi. È importante notare che, durante questo periodo di transizione, i nodi federati non ottengono alcun guadagno dall'inserimento di blocchi, a differenza dei pool.

Ad ogni epoca il parametro è gradualmente diminuito, fino alle 22:44 BST del 31 marzo, in cui la community di Cardano ha festeggiato il $d = 0$. La produzione dei blocchi è diventata completamente decentralizzata e a questa fase seguirà la decentralizzazione della governance e della rete.

3.1.5 *Ouroboros: PoS o DPoS?*

Il meccanismo di delega introdotto in Ouroboros, grazie al quale gli utenti forniscono a un pool maggiori probabilità di "vittoria", potrebbe indurre a pensare che il protocollo rientri nella categoria Delegateted Proof of Stake (DPoS), tuttavia non è così per due motivi:

1. in un sistema DPoS, le parti interessate votano chi è responsabile della produzione dei blocchi. È diverso dal PoS Cardano, dove le stakes sono delegate a stake pools piuttosto che utilizzate come meccanismo di voto. Un maggior numero di deleghe in un pool non determina in modo certo la sua vittoria, in quanto la selezione avviene comunque randomicamente, come nella categoria di protocolli PoS, di fatto aumenta soltanto la probabilità di vittoria.

2. I protocolli DPoS si basano su una quantità fissa di delegati da votare, il che significa che c'è una quantità stabilita di partecipanti autorizzati a far avanzare la rete, a differenza di Cardano in cui ogni utente potenzialmente può proporsi come pool ed essere selezionato.

In definitiva, alla luce di questi aspetti, notiamo come Ouroboros sia più vicino ai concetti chiave dei protocolli PoS con l'aggiunta del meccanismo di delega, piuttosto che ai protocolli DPoS.

Capitolo 4

Il futuro di Cardano

4.1 *Era Goguen*

Con l'era Goguen, Cardano introdurrà gli smart contracts dando così la possibilità di creare applicazioni decentralizzate, le cosiddette DApps. Non a caso questa era prende il nome da un importante informatico statunitense. L'arrivo di questa funzionalità sull'ecosistema coincide con l'hard fork Alonzo, prevista entro la fine del Q2 2021.

4.1.1 *Struttura*

Cardano si divide in due layers:

- Cardano Settlement Layer (CSL) dove si collocano tutte le informazioni sulle transazioni, similmente a Bitcoin (importo, mittente, destinatario, data del trasferimento). Su questo livello vengono trasferiti gli ADA;
- Cardano Control Layer (CCL) che gestisce tutta la parte relativa al controllo e alle condizioni delle transazioni, quindi gli smart contracts.

La divisione è giustificata da una maggior semplicità di gestione e modifica da parte degli sviluppatori e da una maggior sicurezza: un' eventuale compromissione di uno dei due layer non influisce sull'altro. Le applicazioni andranno a collocarsi, quindi, nel CCL e saranno composte da due parti:

- on-chain: la parte di codice dello smart contract che viene caricata sulla blockchain;
- off-chain: la parte di codice che gira sul client dell'utente.

4.1.2 *Plutus*

Cardano, per mezzo della sua piattaforma Plutus, consentirà agli sviluppatori di utilizzare un solo linguaggio di programmazione per entrambe le parti (off e on chain): Haskell.

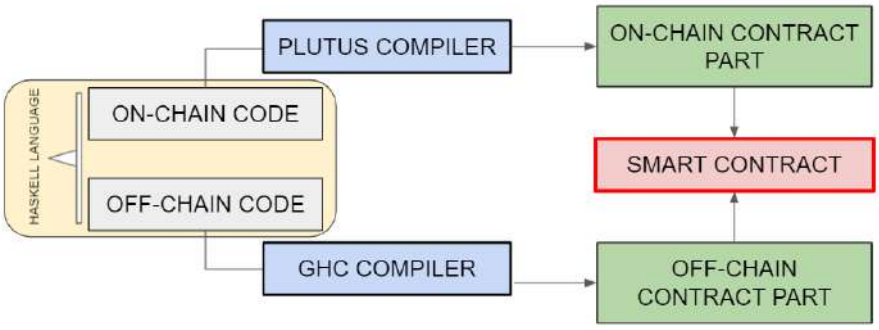


Figura 4.1: Struttura della piattaforma Plutus

Haskell è un linguaggio di programmazione funzionale in cui viene incoraggiata la creazione del proprio sistema utilizzando funzioni pure; è un linguaggio sufficientemente diffuso e, grazie al suo paradigma strettamente funzionale, garantisce determinismo e un alto grado di verifica formale, caratteristiche fondamentali per uno sviluppatore che deve andare a scrivere un codice destinato a funzioni finanziarie. Trascurare questi aspetti può portare a falle di sicurezza con conseguenze veramente gravi, direttamente proporzionali alla capitalizzazione di mercato della moneta (vedi caso DAO di ETH). Inoltre, le avanzate funzionalità di Haskell consentono di impiegare un'intera gamma di potenti metodi per garantire la correttezza del codice, come basare l'implementazione su specifiche formali ed eseguibili, test estesi basati su proprietà e l'esecuzione di test in simulazione.

4.1.3 *Marlowe*

Un'altra innovazione di Cardano è Marlowe. Marlowe è un DSL (Domain Specific Language) creato appositamente per la finanza ed è utilizzabile anche da chi non ha esperienza nel campo della programmazione, infatti si pone ad un livello più alto di Haskell. Utilizzando questo linguaggio i vantaggi sono molteplici:

- restringe la "zona di operabilità" del programmatore, limitando il rischio di creare un codice che potrebbe essere vulnerabile;
- consente di avere un controllo immediato sul programma e sulla sua logica finanziaria;
- dato il dominio ristretto di keywords è possibile implementare dei tool per agevolare lo sviluppo, ad esempio software di emulazione.

Per agevolare ancora di pù la programmazione per i non-programmatori, è possibile utilizzare l'interfaccia grafica Marlowe Playground, che consente di creare

smart contracts mediante la concatenazione di blocchi (in stile Scratch). Anche in questo caso vige una tutela da parte del sistema stesso.
Lo scopo di Marlowe è riuscire a raggiungere la maggior utenza possibile alla piattaforma garantendo semplicità ed affidabilità.

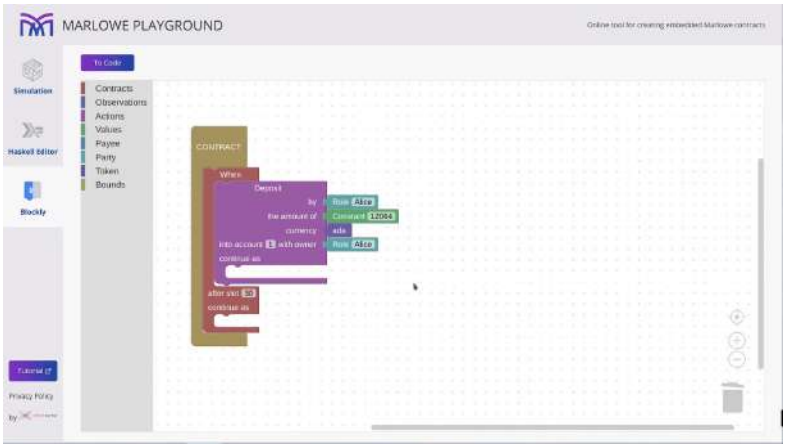


Figura 4.2: Interfaccia grafica di Marlowe playground

4.1.4 *Token nativi*

Goguen darà la possibilità di creare token nativi, cioè token non gestiti da smart contracts, come invece accade su altre blockchain, con i relativi problemi di tasse che questo comporta, vedi Ethereum. I token nativi che verranno creati dagli utenti saranno gestiti allo stesso modo di ADA sulla blockchain Cardano, quindi sul ledger, e ognuno di questi verrà identificato con un codice (fingerprint).

4.2 *Era Basho*

L'era Basho prende il nome dal poeta giapponese del XVII secolo Matsuo Bashō ed è l'era che ambisce ad aumentare la scalabilità e l'interoperabilità della rete.

4.2.1 *Sidechain e scalabilità*

Cardano mira ad aumentare scalabilità e interoperabilità della rete grazie alla possibilità di affiancare alla blockchain principale delle sidechain, le quali permetteranno di espandere l'ecosistema Cardano e di interfacciarsi con altre importanti blockchain.
Un'ulteriore versatilità dell'ecosistema sarà portata dal fatto che queste sidechains non dovranno essere per forza basate su un modello UTXO, ma anche su un modello più classico basato su account. Le sidechain, inoltre, sono un ottimo ambiente in cui gli sviluppatori possono effettuare esperimenti sicuri prima di implementare nuove funzionalità.

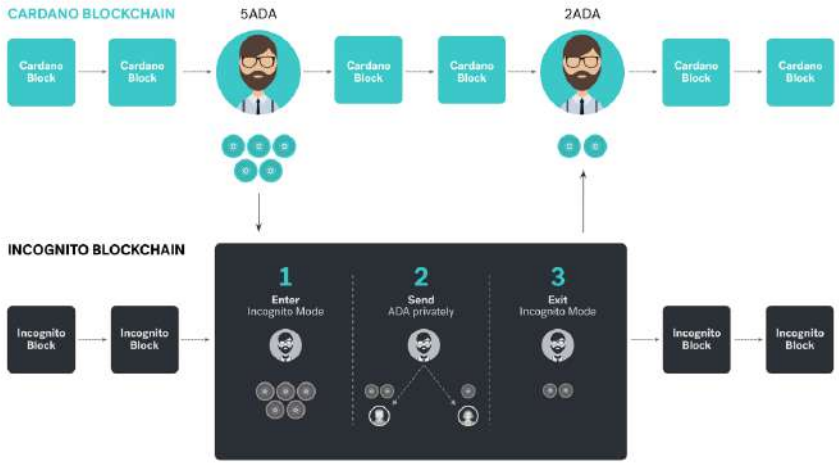


Figura 4.3: Esempio: in questo caso la sidechain in questione offrirebbe un servizio di maggiore privacy agli utenti Cardano. Gli ADA vengono bloccati sulla chain principale e resi disponibili sulla side; l'utente ne fa l'uso che vuole su quest'ultima e poi può ritrasferire il saldo che ha ottenuto dalle eventuali transazioni sulla chain principale.

Un'altra tecnica per aumentare la scalabilità della rete è la suddivisione della blockchain in frazioni: un nodo non dovrà più contenere l'intera lista dello storico di transazioni, ma solo una parte.

4.3 *Era Voltaire*

L'era Voltaire è l'ultimo step della roadmap di Cardano e prende il nome dal noto filosofo francese esponente del movimento illuminista. Questa era si pone lo scopo di raggiungere una totale decentralizzazione.

4.3.1 *Governance decentralizzata*

La governance decentralizzata avrà due fondamenti:

- un fondo per lo sviluppo e la manutenzione che verrà finanziato dal 5% delle tasse di transazione accumulate in ogni epoca. Servirà a finanziare eventuali manutenzioni e/o modifiche al sistema.
- Delle votazioni: le modifiche appena citate verranno decise tramite un sistema di votazione degli stakeholders, così da far diventare Cardano una specie di blockchain democratica.

Grazie a questo sistema Cardano potrà diventare indipendente da qualsiasi istituzione e separarsi dalla IOHK Foundation.

Capitolo 5

Progetti in Africa

Cardano è una delle poche blockchain che mira ad aiutare le nazioni sottosviluppate secondo il principio del *"Banking the Unbanked"*, cioè introducendo un sistema finanziario in paesi in cui esso è assente.
L'Africa è un continente estremamente giovane con un ampio focus sullo sviluppo digitale in cui esistono milioni di portafogli digitali. Con essi gli individui possono effettuare rapidamente transazioni attaverso applicazioni mobili piuttosto che con le banconote.

La *"Africa strategy"* è la strategia ideata per spingere gli utenti e le imprese del continente africano ad adottare Cardano. Questa strategia richiede:

- il coinvolgimento degli stakeholders locali nella risoluzione dei problemi reali del mercato: i governi, le organizzazioni non governative (ONG) e le compagnie del settore privato riconoscono il potenziale della tecnologia blockchain per migliorare la fiducia, la trasparenza e l'efficienza;
- la formazione di sviluppatori locali per creare soluzioni ai problemi di quelle aree.

5.1 *Memorandum of Understanding*

Il Memorandum of Understanding (MOU), firmato da IOHK e dal Ministero della Scienza e della Tecnologia etiope il 3 Maggio 2018, ha due obiettivi:

1. formare e assumere giovani sviluppatori di software;
2. utilizzare la blockchain Cardano nell'industria agricola.

Per quanto riguarda il primo obiettivo, IOHK ha offerto un corso sul linguaggio Haskell per circa 100 programmatori etiopi: è stato esplicitamente richiesto dal Ministro che il corso venisse frequentato, inizialmente, da sole donne, in modo tale da promuovere e sottolineare l'importanza e la partecipazione delle donne

nella programmazione.

In ambito agricolo, il Ministro ha affermato di voler utilizzare la tecnologia blockchain come metodo di tracciamento dell'esportazione dei prodotti agricoli, in particolare del caffè. L'idea è che la tecnologia blockchain permetta a tutti i partecipanti della catena di mercato di tracciare il caffè, mentre questo viaggia dalle zone rurali alle zone commerciali. Una volta inseriti i dati nella blockchain, il compratore può conoscere con esattezza la provenienza del caffè e, per esempio, quali pesticidi siano stati utilizzati nella produzione.

5.2 *Atala PRISM*

L'Etiopia è un chiaro esempio del distacco tra gli ambiziosi obiettivi della moderna tecnologia e le reali circostanze del territorio. A tal proposito, IOHK ha annunciato la partnership con il Ministero dell'Educazione etiope per creare l'applicazione Atala PRISM, un sistema basato sulla blockchain Cardano. Tale applicazione fornisce identità digitali e credenziali controllate solo dagli utenti, che possono essere condivise facilmente, in maniera sicura e privata. Le credenziali possono essere verificate instantaneamente senza la necessità della partecipazione di terzi.

Attraverso Atala PRISM, ogni utente ha la possibilità di gestire diversi documenti relativi a molteplici aree di interesse, come l'amministrazione pubblica e privata, la finanza, la sanità e l'educazione. Proprio in relazione a quest'ultimo ambito, creando un'identità digitale per 5 milioni di studenti e 750 mila insegnanti, il Ministero dell'Educazione etiope ha la possibilità sia di valutare la performance degli studenti sia di esaminare l'eventuale successo di un metodo di insegnamento rispetto a un altro.

"È un paese difficile, quindi se ce la faremo qui, potremo farcela ovunque", ha detto John O'Connor, direttore delle Operazioni in Africa di IOHK. Al momento l'Etiopia è il paese che utilizza maggiormente la tecnologia blockchain in ambito educativo.

Bibliografia

DEDALUS WALLET, 2015-2021. URL https://daedaluswallet.io/en/.

YOROI wallet, 2018-2021. URL https://yoroi-wallet.com/#/about.

Explaining Cardano's Proof-of-Stake (PoS) vs. Delegated Proof-of-Stake (DPoS) Blockchain, February 2020. URL https://emurgo.io/en/blog/explain-proof-of-stake-pos-dpos.

EdDSA, May 2021. URL https://en.wikipedia.org/wiki/EdDSA#Ed25519.

CardanofortheWorld. Marlowe Playground 2020 Cardano Virtual Summit, July 2020. URL https://youtu.be/dd5R64zud-w?t=744.

Amy Castor. "WHERE COFFEE JUST GROWS": CONNECTING ETHIOPIAN AGRITECH TO THE BLOCKCHAIN, May 2018. URL https://bitcoinmagazine.com/culture/where-coffee-just-grows-connecting-ethiopian-agritech-blockchain.

Amy Castor. Introducing Marlowe, 2020. URL https://alpha.marlowe.iohkdev.io/doc/marlowe/tutorials/introducing-marlowe.html#introducing-marlowe.

Kieran Costello. From Classic to Hydra: the implementations of Ouroboros explained, March 2020. URL https://iohk.io/en/blog/posts/2020/03/23/from-classic-to-hydra-the-implementations-of-ouroboros-explained/.

Criptovalute24. Cardano [ADA] Criptovaluta: cos'è e come funziona. URL https://www.criptovalute24.com/cardano-ada/.

Charles Hoskinson. How we will launch Shelley, April 2020. URL https://www.youtube.com/watch?v=g7uySEgt06c.

Andrey I Incognito. Incognito Mode for Cardano (ADA), February 2020. URL https://we.incognito.org/t/incognito-mode-for-cardano-ada/269.

IOHK. The Plutus Platform, 2020. URL https://playground.plutus.iohkdev.io/tutorial/index.html.

IOHK. Cardano, 2020. URL https://cardano.org/.

IOHK. Blockchain comes of age in Africa, 2021. URL https://africa.cardano.org.

Jake. Daedalus wallet and Yoroi wallet overview, January 2021. URL https://iohk.zendesk.com/hc/en-us/articles/360026058573-Daedalus-wallet-and-Yoroi-wallet-overview.

Simon Josefsson and Ilari Liusvaara. Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032, January 2017. URL https://rfc-editor.org/rfc/rfc8032.txt.

Philipp Kant. Hard Fork combinator: full tech demo, June 2020. URL https://www.youtube.com/watch?v=lGCB7f095UM.

Patryk Karter. Cardano: la produzione dei blocchi ora è 100 % decentralizzata, April 2021. URL https://cryptonomist.ch/2021/04/01/cardano-produzione-blocchi-decentralizzata/.

Svetlin Nakov. Practical Cryptography for Developers, November 2018. URL https://cryptobook.nakov.com/digital-signatures/eddsa-and-ed25519.

John O'Connor. Vision for blockchain in Africa is becoming a reality, May 2018. URL https://iohk.io/en/blog/posts/2018/05/29/vision-for-blockchain-in-africa-is-becoming-a-reality/.

Tomáš Sušánka. Cryptography behind top 20 cryptocurrencies. URL https://www.susanka.eu/coins-crypto/.

Tiziano Tridico. FOCUS su CARDANO ed ADA, March 2021. URL https://www.youtube.com/watch?v=GJn_-m7B7ms.

Werner Vermaak. A Deep Dive Into Cardano, February 2021. URL https://coinmarketcap.com/alexandria/article/a-deep-dive-into-cardano.

Wunderbaer. Parameter "d" in cardano, August 2020. URL https://cardanojournal.com/parameter-d-in-cardano-74.