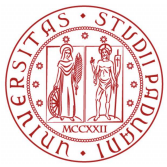


Università degli Studi di Padova

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in
Ingegneria dell'Informazione

L'attacco Parasite Chain per registri distribuiti con struttura Tangle: analisi e contromisure

Relatore:
Prof. Nicola Laurenti

Laureando:
Simone Conton
2010937

Anno Accademico 2022/2023

Abstract

Questa tesi ha l'obiettivo di analizzare gli effetti dell'attacco Parasite Chain in Tangle, una struttura dati a registro distribuito, e di studiare diverse contromisure per prevenirlo. In particolare, verrà introdotta tale tipologia d'attacco, descrivendone una possibile implementazione in Tangle ed evidenziando i rischi derivanti da un eventuale successo. Nello specifico, si provvederà a fornire un modello matematico dell'attacco e si individueranno delle strategie per rendere robusto il sistema di fronte a tali minacce. Infine verranno presentate le soluzioni industriali adottate da IOTA, una criptovaluta che utilizza Tangle, per prevenire e contrastare tale vettore d'attacco.

Indice

Capitolo 1

Tangle

L'avvento delle critpovalute ha contribuito alla diffusione a livello globale dei sistemi a registro distribuito. L'esempio principe di tale tecnologia (abbreviata in **DLT**, da "distributed ledger technology") è sicuramente la Blockchain. Questa struttura dati garantisce la sicurezza dei propri utenti attraverso l'uso di funzioni di hash crittografiche, ed è pertanto usata, nell'ambito delle criptovalute, come luogo in cui memorizzare le transazioni, senza il rischio che queste vengano modificate oppure eliminate. La Blockchain, tuttavia, presenta diversi aspetti negativi, o migliorabili, tra cui la *scalabilità*, la *velocità* e i *costi* delle transazioni.

1.1 Un'evoluzione della Blockchain

In risposta alle criticità della Blockchain è nato *Tangle*, utilizzato da una criptovaluta dedicata al mondo dell'industria dell'IoT, chiamata IOTA.
L'aspetto rivoluzionario di Tangle consiste nell'abbandono della semplice struttura della catena in favore di un grafo aciclico diretto (**DAG**, "direct acyclic graph").
In particolare, Tangle può essere definito come un grafo $G = (V, E)$ con le seguenti proprietà:

- G è finito e due vertici possono essere collegati al più da due archi.
- esiste un vertice speciale g, chiamato genesi, da cui non parte alcun cammino orientato.
- per ogni vertice $v \in V$, tale che $v \neq g$, esiste un cammino orientato da v a g.

1.1.1 Nuove transazioni e approvazione

Affinché un nodo, ovvero un utente, possa immettere in Tangle una transazione è necessario che prima validi m transazioni già presenti (in IOTA $m = 2$), con una modalità simile alla risoluzione del puzzle crittografico di Bitcoin. Con questo meccanismo i nodi che utilizzano la struttura dati lavorano attivamente per garantirne e aumentarne la sicurezza: una transazione validata molte volte è da considerarsi affidabile. Un'interessante conseguenza derivante da questa scelta di implementazione è l'eliminazione del concetto di "miner", presente in altre criptovalute, e la conseguente abolizione delle "tasse" legate all'immissione di una transazione nella struttura dati. In Bitcoin, ad esempio, se un utente vuole registrare una transazione solitamente deve affidarsi ad un "miner", ovvero un utente (o a un gruppo di utenti, o addirittura a un'azienda specializzata) che disponga di una potenza computazionale sufficiente per risolvere il problema crittografico (conosciuto per l'appunto come "puzzle crittografico"): naturalmente il miner viene ricompensato con una tassa, ovvero una quantità di denaro per il lavoro svolto. Poiché il Tangle è nato per essere utilizzato nel mondo dell'IoT, in cui il numero di transazioni è elevato ma la quantità di denaro utilizzata per ogni transazione non è ingente, non avrebbe senso mantenere un sistema di tassazione: si potrebbe addirittura arrivare al paradosso che la tassa superi di valore la transazione stessa.
Inoltre, la scelta di sostituire la catena con un grafo, e quindi la scelta di rendere la struttura dati più complessa viene ripagata con la risoluzione dei problemi di scalabilità che contraddistinguono la Blockchain.
Facendo riferimento alla struttura del grafo (in cui le transazioni corrispondono ai vertici), se tra una transazione y e una transazione x sono collegate da un arco, allora y *approva direttamente* x (in simboli $y \to x$), mentre se tra le due è presente un cammino composto da almeno due archi si afferma che y *approva indirettamente* x ($y \rightsquigarrow x$).

1.1.2 Peso proprio e peso cumulativo

Una metrica fondamentale di Tangle è il *peso proprio* (indicato con $\mathcal{W}$) di una transazione, proporzionale alla quantità di lavoro impiegata per immetterlo nella struttura dati. In IOTA tale quantità assume come valori 3^n, con n limitato, tuttavia non è restrittivo assumere che il peso di ogni transazione sia 1.

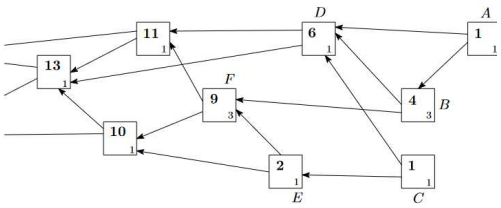
Si definisce inoltre il *peso cumulativo* come:

$$\mathcal{H}_x = \mathcal{W}_x + \sum_{\substack{y_i \to x \\ y_i \rightsquigarrow x}} \mathcal{W}_{y_i}$$

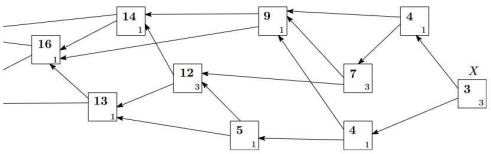
In parole, il peso cumulativo di una transazione è la somma del peso proprio e di tutti i pesi delle transazioni che la approvano, direttamente o indirettamente. Se si assume, come detto in precedenza, che ogni vertice abbia peso proprio unitario, il peso cumulativo di una transazione equivale al numero di transazioni che la approvano, direttamente o indirettamente. L'importanza del peso come parametro risiede nel fatto che una transazione approvata molte volte, e quindi con un peso alto, viene considerata affidabile.

1.2 Modello di crescita

Come già anticipato, la struttura di Tangle garantisce la risoluzione dei problemi di scalabilità che la Blockchain presenta, in virtù del fatto che possono essere immesse diverse transazioni contemporaneamente. Le ultime transazioni ad essere immesse, ovvero tutte quelle che devono ancora essere validate almeno una volta, vengono chiamate *tip*.



(a) A e C sono dei tip: non sono ancora state approvate.



(b) X approva A e C: X è un tip, A e C non lo sono più.

1.2.1 Numero di tip

È di fondamentale importanza che, per ogni istante di tempo t, il numero di tip rimanga costante: se così non fosse ci sarebbero molte transazioni lasciate senza approvazione. Il numero di nuovi tip per unità di tempo può essere modellato attraverso un processo di Poisson $L(t)$ di parametro λ, per semplicità considerato costante nel tempo.
Si supponga che, al momento dell'immissione di una nuova transazione, un nodo

osservi lo stato che aveva Tangle h unità di tempo prima: pertanto, una transazione immessa nell'istante t sarà visibile solo nell'istante $t+h$. Quest'osservazione è dovuta al fatto che l'immissione non sia istantanea, bensì richieda un certo tempo per avvenire correttamente (si pensi ad esempio al fatto che la validazione di altre due transazioni necessiti di un lavoro di validazione, e quindi di tempo).
Da questo consegue che nell'intervallo di tempo $[t-h, t)$ ci saranno λh tip immessi ma non ancora visibili. Considerando anche gli l tip che nel frattempo non sono stati immessi nel Tangle, ovvero quelli che non sono stati ancora validati ma rimangono in attesa, il numero totale corrisponde a $L_0 = l + \lambda h$. Con un ragionamento analogo, supponendo la stazionarietà del processo, durante l'intervallo $[t-h, t)$ ci sono λh vertici che erano tip nell'istante $t-h$ ma ora non lo sono più (perché sono stati validati e sono diventati transazioni a tutti gli effetti), tuttavia il nodo non lo può sapere, visto che osserva lo stato di Tangle di al tempo $t-h$. Quindi, quando il nodo immette la nuova transazione, la probabilità che scelga effettivamente un tip corrisponde a $\frac{l}{l+\lambda h}$ e dunque il numero medio di tip scelti è $\frac{2l}{l+\lambda h}$. Poiché l'arrivo di una nuova transazione **non dovrebbe cambiare il numero medio di tip**, otteniamo l'equazione $\frac{2l}{l+\lambda h} = 1$, che porge $l = \lambda h$, da cui si ricava:

$$L_0 = 2\lambda h$$

Con questo risultato si dimostra che il grafo è effettivamente finito, ovvero che $|V|$ rimane limitata per $t \to \infty$.

1.2.2 Low e High Load Regime

In ultima istanza è possibile differenziare due regimi di crescita di Tangle:

- Low Load Regime: il numero di tip è molto basso. È possibile che ciò sia dovuto al basso flusso di transazioni, oppure al fatto che la latenza del sistema sia molto alta a fronte di dispositivi molto veloci ad immettere transazioni.

- High Load Regime: il numero di tip è alto. In questo caso il flusso di transazioni è elevato e la latenza del sistema combinata al tempo necessario al calcolo fa sì che ci siano numerosi nuovi tip.

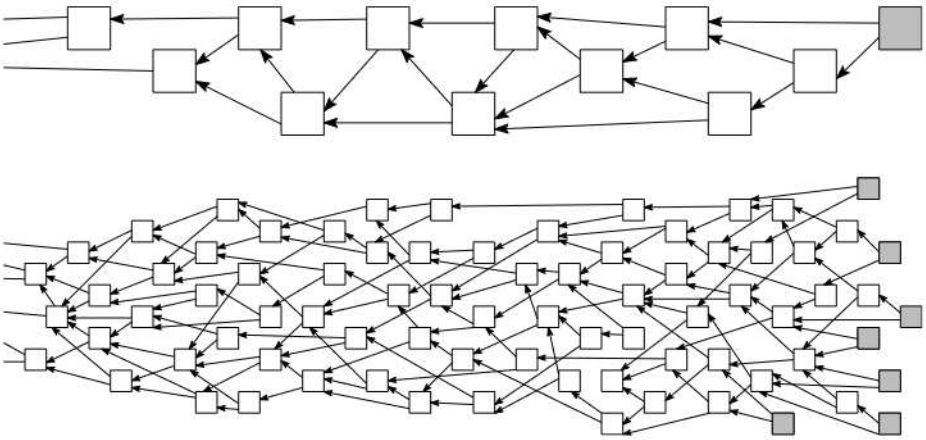


Figura 1.2: In alto un esempio di Low Load Regime, in basso un esempio di High Load Regime.

1.3 Algoritmo di selezione dei tip

1.3.1 Consenso

La mancanza di un'unità centrale che sia in grado di prendere decisioni autonome rende il consenso uno dei problemi più importanti nei registri distribuiti. In particolare, se vengono immessi nella struttura dati due blocchi in conflitto tra di loro, è necessario che i nodi giungano ad un consenso sull'azione da intraprendere, ovvero quale blocco debba essere accettato e quale scartato. Questa situazione si verifica perché non esiste alcun utente privilegiato che possa prendere decisioni. Nel caso delle criptovalute, che utilizzano la struttura dati per memorizzare le transazioni, un esempio di blocchi in conflitto è la situazione in cui si vogliono acquistare due beni attraverso gli stessi fondi. Tale preoccupazione non esiste per un sistema centralizzato, in cui un'entità preposta, come una banca, farebbe da garante per evitare tentativi di spesa attraverso fondi che, di fatto, non esistono. Facendo riferimento al Tangle, questa situazione si presenta al momento della selezione dei tip da accettare. In altre parole, dato l'insieme dei tip, gli utenti del sistema devono decidere quali di essi debbano essere ammessi nel sistema attraverso le validazioni provenienti dalle nuove transazioni in arrivo.
In virtù del fatto che il Tangle è decentralizzato, non può esistere un meccanismo di selezione dei tip che lavori in maniera deterministica, ovvero che scelga a priori quali delle nuove transazioni possano effettivamente fare parte del sistema e quali no. In altre parole, non esistono regole di precedenza per certi utenti piuttosto che per altri. A livello implementativo, quindi, la scelta dei tip dovrà essere **aleatoria**. Da ciò consegue che anche il Tangle è governato dall'aleatorietà, visto

che non segue un'evoluzione deterministica, ma la sua crescita dipende da scelte aleatorie.

1.3.2 URTS

La soluzione più semplice, ma la più rischiosa, consiste nell'adottare un algoritmo di selezione che operi seguendo un modello probabilistico uniforme (denominato **URTS**, da "Uniform Random Tip Selection"), nel senso che ogni tip ha egual probabilità di essere scelto. Naturalmente un approccio di questo tipo è di facile implementazione e porta numerosi vantaggi per quanto riguarda l'efficienza dell'intero sistema, ma i rischi che ne derivano sono troppo importanti per essere trascurati. Ad un ipotetico attaccante infatti basterebbe immettere molte transazioni conflittuali e "sperare" che alcune di queste vengano validate per ottenere un guadagno.

1.3.3 MCMC

Un algoritmo più raffinato fa uso del peso associato ad ogni transazione, basandosi sul metodo Monte Carlo applicato alle catene di Markov (MCMC, da "Monte Carlo Markov Chain").
Una catena di Markov (qui intesa a tempo discreto) è un processo aleatorio $X(nT)$ che soddisfa la proprietà di Markov, ovvero che per ogni intero k, per ogni sequenza ordinata di istanti $n_0 < n_1 < ... < n_k$ e per ogni insieme $\{\sigma_0, \sigma_1, ..., \sigma_k\}$, con $\sigma_i \in \{0, 1, 2...\}$, vale che

$$\mathbb{P}[X_{n_k} = \sigma_k | X_{n_k-1} = \sigma_{k-1}, ..., X_{n_0} = \sigma_0] = \mathbb{P}[X_{n_k} = \sigma_k | X_{n_k-1} = \sigma_{k-1}]$$

Se si considera n_{k-1} il tempo presente e n_k il tempo futuro, allora lo stato futuro del sistema σ_k è determinato unicamente da σ_{k-1}, ovvero dallo stato attuale del sistema.
Con metodo Monte Carlo, inoltre, ci si riferisce all'utilizzo di simulazioni ripetute per ottenere risultati numerici. A lato pratico, l'algoritmo MCMC, che verrà approfondito e studiato in seguito, partendo da una certa profondità del Tangle, si muove verso i tip scegliendo il proprio percorso basandosi sul modello di una catena di Markov (eventualmente pesato da un termine di bias detto α) in cui la probabilità è legata al peso di ogni transazione. Se al termine di un elevato numero di esecuzioni dell'algoritmo un certo tip verrà scelto con il metodo Monte Carlo il 98% delle volte, allora quel tip avrà il 98% di fiducia da parte del sistema.

Tuttavia, è prevedibile che l'utilizzo di tale algoritmo e le numerose esecuzioni richieste dal metodo Monte Carlo costituiscano un problema per quanto riguarda l'efficienza dell'intero sistema.

Capitolo 2

Attacco Parasite Chain

2.1 Double Spending

La situazione in cui vengono utilizzati gli stessi fondi per compiere due diverse transazioni è nota, nel mondo delle criptovalute, come "double spending". Nel caso di un sistema centralizzato, esiste un'unità centrale che fa da garante affinché non si verifichi mai. Per quanto riguarda i sistemi decentralizzati, invece, il double spending rappresenta una seria minaccia, poiché non esiste alcuna entità privilegiata che abbia la facoltà di approvare o scartare una transazione.

2.1.1 Blockchain e potenza computazionale

Facendo riferimento, ad esempio, alla classica struttura della Blockchain, è possibile che vengano aggiunti alla catena, contemporaneamente, due blocchi diversi. Gli utenti in questo caso non sanno, in virtù della completa uguaglianza che vige nel sistema, quale dei due blocchi accettare. In Bitcoin, è stato deciso che i nodi in caso di biforcazione scelgano sempre la catena più lunga, scartando l'altra. Un attaccante potrebbe sfruttare questo sistema di decisione provando a costruire più rapidamente una catena che superi quella onesta. Se questo accade, egli è in grado di risalire la catena fino ad invalidare una transazione precedente dopo averne tratto dei benefici (come l'acquisto di beni), realizzando così il double spending.
In quest'ottica entra in gioco la potenza computazionale dell'attaccante, nel senso che se esso dispone di più della metà della potenza totale dell'intero sistema, allora *asintoticamente* avrà la meglio. A questo proposito è doveroso considerare anche le tempistiche per portare a termine l'attacco: un eventuale attaccante non

potrebbe accontentarsi di realizzare il double spending in tempi molto dilatati, pertanto dovrebbe aumentare la propria potenza computazionale per cercare di avere la meglio il prima possibile.
D'altro canto, non è necessario disporre di almeno la metà della potenza dell'intero network se si vuole che il proprio attacco vada a buon fine: la probabilità di successo infatti non è nulla, pertanto anche un avversario che dispone di un notevole quantitativo di risorse, ma non necessariamente la maggioranza, rappresenta una grave minaccia. A livello asintotico, tuttavia, in questo caso, egli non avrà successo.
Se si modella la costruzione della nuova catena come una binomiale, con
$\mathbb{P}$(il prossimo blocco è trovato da un nodo onesto) = p
$\mathbb{P}$(il prossimo blocco è trovato dall'attaccante) = q
$\mathbb{P}$(l'attaccante recupera lo svantaggio di z blocchi) = q_z
Segue che

$$q_z = \begin{cases} 1 & \text{se } p \leq q \\ (\frac{q}{p})^z & \text{se } p > q \end{cases}$$

Nel caso in cui $p \leq q$, come detto in precedenza, l'attacco ha successo asintoticamente, visto che, anche qualora la catena onesta riuscisse ad aggiungere un blocco, l'attaccante riuscirebbe a "superarla" grazie alla maggiore potenza di calcolo.
Al contrario, se non dispone di più della metà della potenza di calcolo, e quindi $p > q$, le probabilità di successo dell'attacco diminuiscono esponenzialmente all'aumentare di z.

2.2 Implementazione nel Tangle

Nonostante il Tangle abbia una struttura più complessa della semplice catena, è comunque possibile individuare una strategia d'attacco simile a quella appena descritta. Nel Tangle, infatti, viene immessa più di una transazione contemporaneamente. Inoltre, gli algoritmi di selezione dei tip si basano sul peso di ogni vertice del grafo, e quindi considerano anche le transizioni approvate indirettamente.
L'attacco Parasite Chain è composto da tre fasi:

1. all'istante T_0 l'attaccante inizia a costruire, in segreto, una catena parallela al Tangle principale.

2. all'istante T_1 inserisce nel Tangle principale una transazione $\mathcal{T}_1$: sarà la transazione ad essere invalidata dal double spending.

3. all'istante T_2 inserisce nel proprio sub-Tangle una transazione $\mathcal{T}_2$ che utilizza gli stessi fondi di $\mathcal{T}_1$ e rivela al sistema il sub-Tangle costruito, che vorrebbe sostituisse il Tangle principale, invalidando così $\mathcal{T}_1$.

Facendo riferimento all'algoritmo di selezione dei tip, l'attaccante può aumentare notevolmente il peso della Parasite Chain (PC) approvando solamente le proprie transazioni. Nel caso di una Simple Parasite Chain (SPC), ad esempio, le nuove transazioni immesse nella Parasite Chain approvano quella precedente e una transazione r nel Tangle principale, facendone incrementare il peso e quindi forzando l'algoritmo a passare proprio per r. Inoltre, al momento dell'attacco egli può aumentare notevolmente il numero di tip facendo in modo che questi approvino le transazioni della catena.

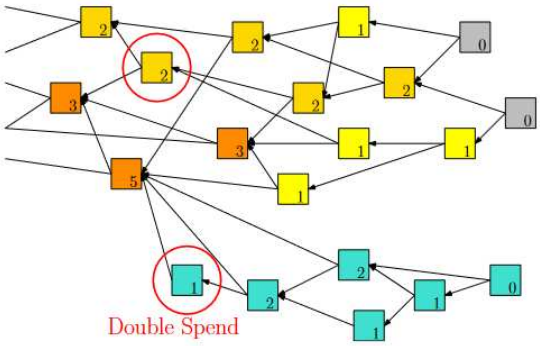


Figura 2.1: Esempio di Parasite Chain: un eventuale attaccante vorrebbe utilizzare gli stessi fondi per una transazione nel Tangle e una nella Parasite Chain, come le due cerchiate nella figura.

2.3 Rilevamento di una Parasite Chain

Una Parasite Chain può avere molteplici forme. Anche facendo riferimento alla Simple Parasite Chain, che rappresenta il caso in cui esiste un unico vertice r di Tangle collegato alla catena, chiamato radice, si possono individuare diversi modi di comporre una Parasite Chain. La forma più semplice consiste in una catena in cui l'i-esimo blocco è collegato all'i-1-esimo e ad r. Al contrario, possono esistere catene composte da alcuni blocchi collegati unicamente alla Parasite Chain. Tuttavia, è di fondamentale importanza ricordare che l'algoritmo di selezione dei tip si basa sul peso di ogni vertice, pertanto catene con forme più "esotiche" ridurranno il peso della radice, rendendo meno probabile che questa venga scelta

durante la selezione dei tip, diminuendo pertanto la probabilità che si scelga la Parasite Chain in favore del Tangle principale. L'obiettivo delle prossime sezioni è quello di ricavare un modello probabilistico del numero di approvazioni dirette che riceve una transazione appartenente al Tangle principale. Si dimostrerà inoltre che una transazione che invece appartiene alla Parasite Chain avrà un numero diverso di approvazioni dirette e si sfrutterà questa differenza per individuare una soglia di confidenza sul numero di approvazioni, che, qualora venisse superata, indicherebbe il rischio che una transazione appartenga alla PC.

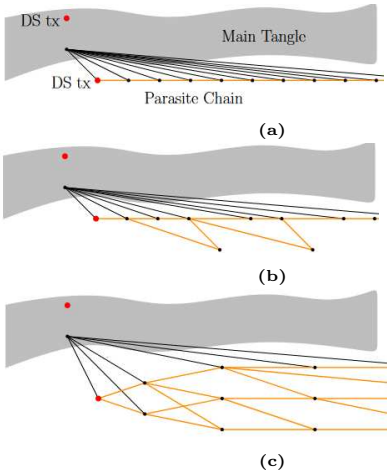


Figura 2.2: La figura **a** rappresenta una PC con un forma estremamente semplice ma molto efficiente, visto che il peso della radice a cui è attaccata nel Tangle è elevato. Le figure **b** e **c** invece prendono in considerazione PC con forme più complesse, ma meno efficienti.

2.3.1 Numero di approvazioni

Come già evidenziato più volte, il numero di numero di approvazioni ricevute da una transazioni indica il suo livello di affidabilità ed è per questo un indice fondamentale nella selezione dei tip. Prima di indicare la probabilità che una transazione selezionata in modo aleatorio abbia un dato numero n di approvazioni, è necessario individuare due scenari relativi alla selezione dei tip:

- Single Edge Model (**SEM**): ad un tip viene collegato un solo arco (ovvero viene validato solo una volta).

- Multi Edge Model (**MEM**): ad un tip vengono collegati più archi (nel caso di IOTA gli archi sono due). Nonostante sia una circostanza poco comune, è comunque possibile. Si pensi ad esempio ad una situazione di *Low Load*

Regime in cui l'algoritmo di selezione dei tip è fortemente sbilanciato in favore di solo uno di essi: la nuova transazione deve compiere due validazioni, pertanto c'è una buona probabilità che validerà due volte lo stesso tip.

Siano p_i la probabilità di essere scelta da parte dell'i-esima transazione, N_i il numero finale di approvazioni, h il tempo necessario per essere visibile al sistema e t_i l'istante della sua prima approvazione. Allora si può provare che:

$$N_i = 1 + p_{i0} + e^{-\lambda_i} \frac{\lambda_i^n}{n}$$

Con

$$p_{i0} = \begin{cases} p_i(t_i) & \text{per MEM} \\ 0 & \text{per SEM} \end{cases},$$

$$\lambda_i = \lambda \int_{t_i}^{t_i+h} dt \begin{cases} 2p_i(t) & \text{per MEM} \\ 2p_i(t) - p_i(t)^2 & \text{per SEM} \end{cases},$$

e λ che rappresenta il rate di transazioni espresso in termini di unità di h. In generale, la probabilità di avere n approvazioni è esprimibile come:

$$P(n) = \sum_{i=1}^{N} \mathbb{P}(N_i = n),$$

dove N è inteso come la cardinalità dell'insieme delle transazioni considerate.
Se si considera l'algoritmo URTS, $p_i = \frac{1}{L}$. Facendo riferimento al fatto che $L = \lambda h$, in *Low Load regime* (ovvero per λ piccolo) il numero dei tip può essere espresso come $L = 1+\lambda h$. La probabilità di aver n approvazioni allora corrisponde alla distribuzione di probabilità di Poisson $P_{URTS}(n)$, che è peraltro uguale alla probabilità di avere $n-1$ approvazioni oltre alla prima, esprimibile con $P(\lambda, n-1)$ di parametro

$$\lambda_U = \begin{cases} \frac{2\lambda}{L} & \text{per MEM} \\ \frac{2\lambda}{L}(1 - \frac{1}{2L}) & \text{per SEM} \end{cases}$$

Vale la pena notare che in *High Load Regime* (ovvero con λ grande) il termine $(1 - \frac{1}{2L})$ è trascurabile, e quindi le distribuzioni di probabilità diventano molto simili. Questo risultato è coerente col fatto che in *High Load Regime* la probabilità che una nuova transazione validi due volte lo stesso tip è decisamente bassa. Passando all'algoritmo MCMC, per un numero L di tip, le probabilità d'uscita variano a seconda del tip. Pertanto è possibile ricavarne una distribuzione,

descritta dalla funzione $e(x)$, normalizzata dall'intervallo $\{1, ..., L\}$ al canonico intervallo $(0, 1]$. Quindi, la probabilità d'uscita normalizzata dell'i-esimo tip è:

$$e_L(i) = \int_{(i-1)/L}^{i/L} e(x)dx,$$

con $e(x) = 1 + f(x)$ e $x \in [0, 1]$, $\int_0^1 f(x)dx = 0$, $f(x) \geq -1$, che sono vincoli necessari data la definizione di $e(x)$. Combinando i risultati analitici alle simulazioni numeriche (su 10^3 campioni di tip, ripetute 10^6 volte seguendo il metodo Monte Carlo) si ottiene un grafico in cui si notano immediatamente le differenze tra i due tipi di approccio.

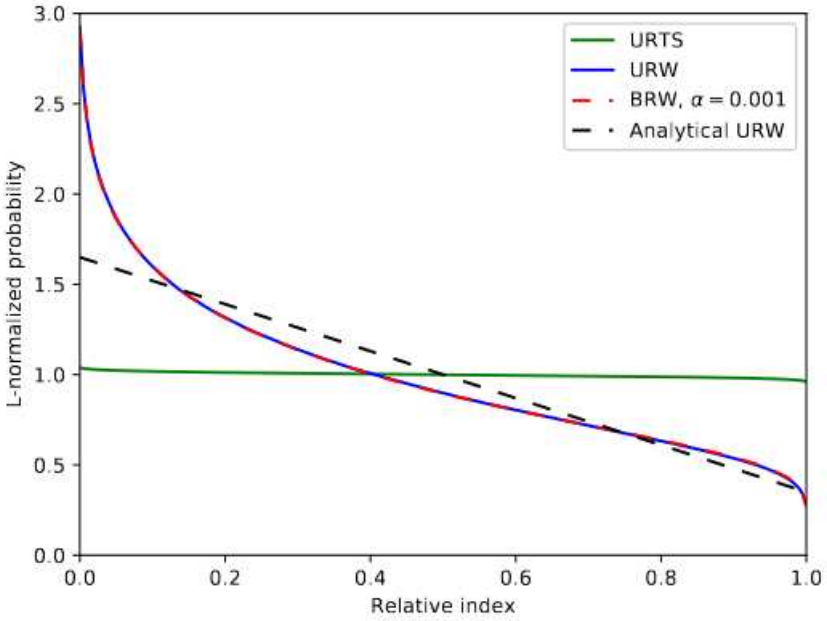


Figura 2.3: Simulazioni numeriche per URTS e MCMC (con e senza bias) della funzione $e(x)$. Tratteggiata in nero si vede già una possibile approssimazione lineare per gli algoritmi MCMC.

Seguendo un approccio simile a quello utilizzato per l'algoritmo URTS, è possibile definire la distribuzione di massa per la probabilità di avere n approvazioni per un tip che ha probabilità d'uscita x, con $x \in (0, 1]$, nel caso si usino gli algoritmi MCMC (in questo caso senza termine di bias):

$$p_{MCMC}(n, x) = P(e(x)\lambda, n - 1)$$

Integrando in dx, si ricava una forma simile a quella dell'algoritmo URTS:

$$P_{MCMC}(n) = \int_0^1 e^{-e(x)\lambda} \frac{(e(x)\lambda)^{n-1}}{(n-1)!} dx$$

È molto interessante osservare che:

$$P_{MCMC}(n) = P_U(n) \int_0^1 e^{-f(x)\lambda} (1 + f(x))^{n-1} dx$$

ovvero, la probabilità di avere n approvazioni da parte di un tip, in base all'algoritmo scelto, sono molto simili, differenziati da un fattore moltiplicativo legato alla funzione $f(x)$.

Inoltre può essere introdotta una semplificazione (approssimazione lineare) scegliendo $f(x) = a(x - 0.5)$, $a \in [0, 2]$, e a scelto in modo che $f(x) > 0$. Ora la probabilità diventa

$$P_{MCMC}(n) = P_{URTS}(n) g(n-1),$$

con

$$g(n) = \frac{1}{a} \sum_{j=0}^{n} \lambda^{-j-1} \frac{n!}{(n-j)!} [e^{-y\cdot\lambda} (1+y)^{(n-j)}]_{-0.5a}^{0.5a}.$$

2.3.2 Distanza rispetto al numero di approvazioni

Il numero di approvazioni in una SPC è in una certa misura diverso, a livello di distribuzioni di massa, rispetto a quello del Tangle principale. Per questo motivo viene introdotta una metrica di distanza sulle distribuzioni: qualora si misurasse una distanza significativa rispetto alle distribuzioni "canoniche" (in particolare quelle derivate precedentemente mediante l'approccio lineare) allora è probabile che ci si trovi di fronte ad una PC. Si definisce quindi come metrica di distanza:

$$d_P = \frac{1}{2} \sum_{n=0}^{\infty} |P(n, S) - P_{ref}(n)|$$

dove $P(n, S)$ rappresenta la distribuzione di massa rispetto ad un numero S di campioni, $P_{ref}(n)$ è la distribuzione ricavata analiticamente e il fattore $\frac{1}{2}$ serve a normalizzare la distanza.
Per un attaccante è molto difficile, e computazionalmente costoso, inserire transazioni con un elevato numero di approvazioni. Per questo motivo, questo tipo di transazioni si troveranno nel Tangle, ma la distanza d_P delle loro distribuzioni

che descrivono la probabilità di avere un certo numero di approvazioni rispetto alle distribuzioni di massa canoniche sarà elevata. La soluzione sarebbe "premiare" tali transazioni, invece di penalizzare: a questo proposito si introduce una seconda metrica:

$$d_Q = \frac{1}{2} \sum_{n=0}^{\infty} |Q(n) - Q_{ref}(n)|$$

in cui si sostituisce la distribuzione di massa $P(n)$ con quella che viene chiamata distribuzione di probabilità cumulativa:

$$Q(n) = \sum_{m=n}^{\infty} P(m)$$

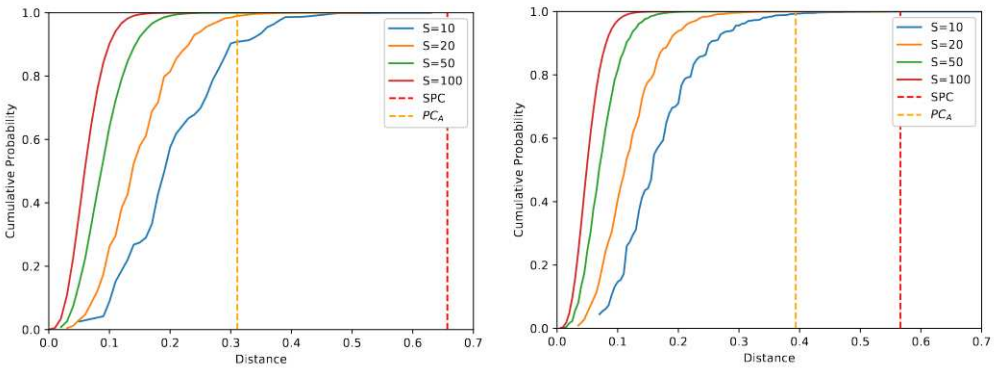


Figura 2.4: La figura mostra la probabilità cumulativa, per diversi campioni S appartenenti al Tangle, che essi abbiano una certa distanza (a destra è stata usata la metrica d_P, a sinistra la metrica d_Q) dalle distribuzioni di riferimento. Le stesse misurazioni sono state effettuate per due modelli di Parasite Chain: è evidente la differenza rispetto al Tangle.

Il fatto di aver ricavato due metriche di distanza relative alle distribuzioni di massa che descrivono il numero di approvazioni per una specifica transazioni permette di definire un meccanismo di rilevamento delle Parasite Chain basato proprio sulla distanza. In particolare, è possibile introdurre un parametro η che rappresenti una soglia entro la quale la distanza d rispetto al modello canonico individuato possa essere ancora considerata "sicura". Naturalmente tutto ciò è possibile in virtù del fatto che la struttura della PC è notevolmente diversa da quella di grafo che ha il Tangle: in particolare, a cambiare è il numero di approvatori diretti.

L'algoritmo di selezione pertanto può essere implementato tenendo conto delle metriche d_P e d_Q in relazione alla soglia η. Nello specifico, se la soglia critica η viene superata, allora è auspicabile interrompere l'esecuzione dell'algoritmo e farlo

ricominciare da capo, evitando così il rischio di incorrere in un attacco Parasite Chain.

Capitolo 3

Analisi dell'attacco PC con algoritmo di selezione MCMC

3.1 Caratteristiche dell'algoritmo

L'algoritmo di selezione dei tip è già stato introdotto precedentemente. Nello specifico, l'algoritmo segue i seguenti passi:

- Scegliere un intervallo da cui far partire le random walk. Si potrebbe partire sempre dalla genesi g, ma quando il Tangle raggiunge dimensioni considerevoli, questa scelta implicherebbe una minore efficienza. L'intervallo, quindi, dev'essere posizionato ad una profondità del Tangle tale per cui ci sia un compromesso tra efficienza ed efficacia (non sarebbe efficace se, ad esempio, partisse molto vicino ai tip).

- Individuare in modo indipendente l'uno dall'altro M siti (vertici) all'interno dell'intervallo: saranno la partenza.

- Eseguire le random walk verso i tip. Per passare da una transazione x ad una transazione y è necessario che $y \to x$.

- I due tip che sono raggiunti per primi saranno quelli scelti da approvare.

È doveroso porre attenzione alla scelta degli M vertici di partenza: IOTA, nel suo protocollo, non specifica in alcun modo il criterio con cui essi debbano essere scelti. Se il criterio è basato su una distribuzione di probabilità uniforme, è possibile che l'algoritmo parta da un vertice a cui è collegata una PC, da cui deriva il rischio di incontrare una Parasite Chain già dalla partenza. In egual

modo, se il criterio invece si basa su transazioni che hanno molto peso, ovvero che sono state approvate molte volte e dunque sono considerate affidabili, c'è il rischio che un eventuale attaccante incrementi notevolmente il peso della radice r per renderlo uno dei siti prescelti. Una scelta affidabile sarebbe indicare la genesi g come punto di partenza delle random walk, tuttavia per Tangle di grandi dimensioni rappresenterebbe un'importante perdita di efficienza, visto che sarebbe necessario attraversare ogni volta l'intero Tangle prima di raggiungere i tip.

L'importanza dei pesi cumulativi per l'algoritmo risiede nella probabilità di transizione P_{xy}, ovvero la probabilità che la random walk, nel proprio percorso, passi dalla transazione x alla transazione y. In particolare:

$$P_{xy} = \frac{f(-\alpha(\mathcal{H}_x - \mathcal{H}_y))}{\sum_{z:z\to x} f(-\alpha(\mathcal{H}_x - \mathcal{H}_z))}$$

dove α è il parametro di bias già menzionato e $f : \mathbb{R}_+ \to \mathbb{R}_+$ è una funzione non decrescente.

Partendo dal presupposto che l'attaccante non disponga della maggioranza della potenza di calcolo totale, si intuisce che l'algoritmo possa essere una contromisura efficace. Infatti, qualora una delle random walk dovesse incontrare un vertice a cui è collegata la PC, la probabilità di transizione in favore del Tangle principale sarebbe maggiore rispetto a quella di cadere nel sub-Tangle dell'attaccante.
In uno scenario invece in cui uno o più avversari detenessero una parte considerevole della totale potenza computazionale, ovvero il caso in cui riuscissero a rendere il peso di una Parasite Chain maggiore di quello del Tangle principale, allora il double spending sarebbe inevitabile a livello asintotico. Questo significa che, qualora riuscissero a mantenere tale vantaggio computazionale per abbastanza tempo, avrebbero la meglio sul sistema.

3.2 Modello basato su una catena di Markov

Per una trattazione più rigorosa, si introducono $\mathcal{L}(t)$, ovvero l'insieme di tutti i tip all'istante t (si noti che $|\mathcal{L}(t)| = L(t)$) e $q \in [0, \frac{1}{2})$, parametro che è legato alla possibilità che la random walk non prosegua verso i tip ma torni indietro verso la genesi g. Allora la probabilità di transizione da un vertice x ad un vertice y è

esprimibile con:

$$P_{xy}^{(t)} = \begin{cases} \frac{q}{|\mathcal{A}(x)|} & \text{se } y \in \mathcal{A}(x) \\ (1-q)\frac{f(-\alpha(\mathcal{H}_x^{(t-h)} - \mathcal{H}_y^{(t-h)})}{\sum_{z:z\in\mathcal{A}(z)} \mathcal{H}_x^{(t-h)} - \mathcal{H}_z^{(t-h)}} & \text{se } x \in \mathcal{A}(y) \\ 0 & \text{altrimenti} \end{cases}$$

con $\mathcal{A}(x)$ che rappresenta l'insieme delle transazioni *approvate da* x, $\mathcal{A}(y)$ l'insieme delle transazioni che *approvano* x, q la probabilità che da x la random walk vada verso un vertice $v \in \mathcal{A}(x)$, ovvero che torni verso la genesi piuttosto che dirigersi verso i tip. È stata inoltre introdotta la dipendenza dal tempo t per quanto riguarda la probabilità di transizione, che però fa riferimento allo stato del Tangle (e quindi ai pesi dei vari vertici) nell'istante $t - h$, visto che l'utente che immette la transazione non osserva lo stato attuale del Tangle.
Un modello matematico adatto per descrivere la random walk su cui si basa l'algoritmo è una catena di Markov composta da due sottocatene, una riferita al Tangle e una alla PC, comunicanti tra loro.

3.2.1 Probabilità di transizione

Per una catena di Markov (MC) è di fondamentale importanza definire cosa siano le probabilità di transizione. In particolare, ci si riferisce agli istanti di tempo come "passi"; inoltre, per una generica MC $X(nT)$, quando $X_n = j$ si intende che la catena è nello stato $j \in \mathcal{S}$ all'n-esimo passo. Il concetto di passo comprende quindi anche il passaggio di un'unità di tempo, infatti, se $X_n = i$ e $X_{n+k} = j$, si dice che la MC $X(nT)$ è passata dallo stato i allo stato stato j in k passi. La probabilità condizionata $\mathbb{P}(X_{n+k} = j | X_n = i)$ è detta probabilità di transizione; se dipende solo da k la MC è detta omogenea.
Riprendendo quanto detto nelle sezioni precedenti, si assume che due transazioni collegate direttamente siano separate da un intervallo temporale pari a $2h$, con h che rappresenta il tempo necessario ad una transazione per essere visibile all'intero sistema. In IOTA ogni transazione ne deve approvare **due** e, in questo caso, **si suppone che venga, in media, approvata da altre due**.
Dopo aver introdotto λ (ovvero il rate di immissione delle transazioni nel Tangle), si definisce μ come il rate di immissione delle transazioni nella PC. Infine, si suppone che a regime il peso cumulativo delle transazioni cresca linearmente con rate λ prima che la PC venga rivelata, e con rate $\lambda + \mu$ dopo, ovvero
$\mathcal{H}_x^{(t)} \approx (\lambda + \mu)(t - t_x) + c$, con la costante $c \in \mathbb{R}$ e t_x che rappresenta l'istante di

immissione della transazione x.
Facendo riferimento ai tre istanti T_0, T_1 e T_2, introdotti in precedenza, che caratterizzano i tre momenti chiave dell'attacco, per modellare il Tangle e la PC è necessario calcolare il numero di stati nelle due sottocatene tra un istante T_i e un istante T_j.
Per quanto riguarda il Tangle principale, ogni "passo" a livello temporale è equivalente a $2h$, pertanto il numero di stati sarà:

$$n_{tangle_{i,j}} = \lceil 1 + (T_i - T_j)/2h \rceil$$

A proposito della Parasite Chain, invece, ogni "passo" a livello temporale è di lunghezza $\frac{1}{\mu}$, visto che gli istanti di arrivo delle transazioni della PC sono modellate con una probabilità esponenziale con media $\frac{1}{\mu}$, quindi:

$$n_{parasite_{i,j}} = \lceil 1 + (T_i - T_j) \cdot \mu \rceil$$

Si definiscano inoltre gli insiemi degli stati in cui si può trovare una transazione come:

$$\mathcal{T} = \{i \in \mathbb{N}_+ |\ 0 \leq i \leq n_{tangle,02}\} \text{ con } |\mathcal{T}| = n_{tangle,02}$$
$$\mathcal{P} = \{i \in \mathbb{N}_+ |\ n_{tangle,02} \leq i \leq n_{tangle,02} + n_{parasite,02}\} \text{ con } |\mathcal{P}| = n_{parasite,02}$$

Si identifichi con x lo stato di una transazione appartenente al Tangle principale, ovvero $x \in \mathcal{T}$. Le probabilità di transizione da tale stato ad uno stato y appartenente al Tangle *oppure* alla PC sono:

$$P_{xy}^{(f)} = \begin{cases} 1 & \text{se } x = n_{tangle,01} - 1 = y \\ q & \text{se } x < n_{tangle,02} \text{ e } x = y + 1 \text{ o } x = y = 0 \\ w_x^{(t)} & \text{se } x \leq n_{tangle,01} \text{ e } x = y - 1 \text{ e } x < n_{tangle,02} - 1 \\ 1 - q & \text{se } x > n_{tangle,01} - 1 \text{ e } x = y + 1 \text{ e } x < n_{tangle,02} - 1 \\ 1 - q - w_x^{(f)} & \text{se } x \leq n_{tangle,01} - 2 \text{ e } y = \text{argmin}\{z \in P |\ t_z \geq t_x\} \\ 1 - q - w_x^{(f)} - \sum_{z \in P: z \neq y} v_z^{(f)} & \text{se } x = n_{tangle,01} - 1 \text{ e } y \geq n_{tangle,02} + n_{parasite,01} - 1 \\ 0 & \text{altrimenti} \end{cases}$$

Si noti che le probabilità di transizione dipendono, naturalmente, da x e y, ovvero dallo stato iniziale e dallo stato finale. È di fondamentale importanza ricordare che gli stati sono stati numerati, e in questo caso x e y sono gli indici

assegnati a ciascuno stato (con $0 \leq x < n_{tangle,02}$ e $0 \leq y < n_{tangle,02} + n_{parasite,02}$). Per ricavare la funzione $w_x^{(f)}$ è opportuno notare che ogni transazione nella SPC deve obbligatoriamente approvare una transazione nel Tangle principale, pertanto la probabilità di essere approvata per una transazione nel Tangle dipende dai parametri λ e μ. Quindi:

$$\mathbb{P}(\text{Una transazione nel Tangle non è approvata da una transazione della PC}) = \frac{\lambda}{\lambda + \mu}$$

Inoltre, se un vertice del Tangle principale è collegato alla Parasite Chain, per quanto riguarda il percorso della random walk:

$$\mathbb{P}(\text{Stare nel Tangle} \mid \text{Il vertice corrente è collegato alla PC}) = \frac{2 \cdot f(2h(\lambda+\mu))}{2 \cdot f(2h(\lambda+\mu)) + f(T_2 - t_x)\lambda}$$

Sfruttando queste considerazioni, con un approccio che segue quello di un albero binomiale, si ottiene:

$$w_x^{(f)} = (1-q) \cdot \Bigg($$

$$\mathbf{1}_{\{x \leq n_{tangle,01}-2\}} \cdot \left(\frac{\lambda}{\lambda+\mu} + \frac{\mu}{\lambda+\mu} + \frac{2 \cdot f(2h(\lambda+\mu))}{2 \cdot f(2h(\lambda+\mu)) + f(T_2 - t_x)\lambda} \right) +$$
$$\mathbf{1}_{\{x = n_{tangle,01}-1\}} \cdot \left(\frac{2h\lambda}{2h\lambda + (T_2 - T_1)\mu} + \frac{(T_2 - T_1)\mu}{2h\lambda + (T_2 - T_1)\mu} + \frac{2 \cdot f(2h(\lambda+\mu))}{2 \cdot f(2h(\lambda+\mu)) + \sum_{z \in \mathcal{P}: t_z \geq t_x} f((T_2 - T_z)\lambda)} \right) \Bigg)$$

Inoltre:

$$v_z^{(f)} = (1-q) \cdot \frac{(T_2 - T_1)\mu}{2\lambda h + (T_2 - T_1)\mu} \cdot \frac{f((T_2 - T_1)\lambda}{2 \cdot f(2h(\lambda+\mu) + \sum_{y \in P: t_y \geq T_1} f((T - 2 - t_z)\lambda)}$$

e

$$t_z = \begin{cases} 2hz & \text{se } z \in \mathcal{T} \\ (z - n_{tangle,02} \cdot \mu^{-1}) & \text{se } z \in \mathcal{P} \end{cases}$$

Passando invece al caso $x \in \mathcal{P}$, ovvero quando la transazione x è nella PC, la

probabilità di transizione ad una transazione y è:

$$P_{xy}^{(f)} = \begin{cases} 1 & \text{se } x = n_{tangle,02} + n_{parasite,02} - 1 = y \\ q & \text{se } x = n_{tangle,02} \text{ e } y = 0 \\ q/2 & \text{se } x_{tangle,02} < x < n_{parasite,02} - 1 \text{ e } x = y + 1 \\ q/2 & \text{se } x_{tangle,02} < x < n_{parasite,02} - 1 \text{ e } \operatorname{argmin}\{z \in \mathcal{T} | t_z > t_x\} \\ 1 - q & \text{se } x < n_{tangle,02} + n_{parasite,02} - 1 \text{ e } x = y + 1 \\ 0 & \text{altrimenti} \end{cases}$$

In questo caso $n_{tangle,02} \leq x < n_{tangle,02} + n_{parasite,02}$ e $0 \leq y < n_{tangle,02} + n_{parasite,02}$.

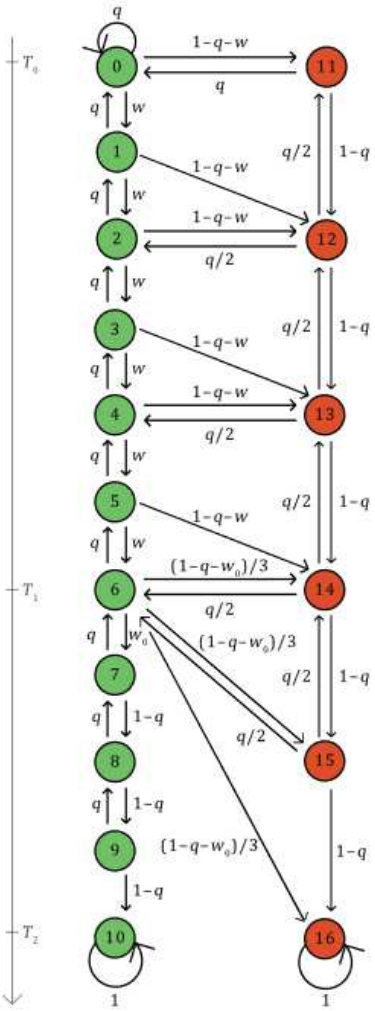


Figura 3.1: Visualizzazione grafica delle probabilità di transizione. Ogni stato del Tangle è stato numerato da 1 a $n_{tangle} = 10$, mentre ogni stato della PC è stato numerato da $n_{tangle} + 1$ a $n_{parasite} = 16$. In questo caso $h = 1$, $\mu = 0.25$, $T_1 - T_0 = 12$ e $T_2 - T_1 = 8$.

3.2.2 Matrice di transizione

Data una catena di Markov, è possibile costruire una matrice quadrata $\mathbf{P}(k)$ dipendente dal tempo in cui all'i-esima riga nella j-esima colonna si trova la probabilità di transizione dallo stato i allo stato j in k istanti.
Dalla teoria delle catene di Markov, uno stato s_i viene definito *assorbente* se è impossibile lasciarlo, ovvero $\mathbb{P}(\text{Rimanere in } s_i) = 1$. Ogni altro stato è detto *transiente*. Una MC è detta *assorbente* se ha almeno uno stato assorbente e tale stato è raggiungibile da qualunque altro stato transiente, non necessariamente con un unico passo.
Per quanto riguarda le probabilità di transizione ricavate precedentemente, la matrice di transizione può essere scritta come:

$$\mathbf{P} = \begin{bmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix}$$

$\mathbf{P}$ è una matrice a blocchi $m \times m$, con $m = n_{tangle,02} + n_{parasite,02}$, $\mathbf{I}_2$ è la matrice identità di dimensione 2, $\mathbf{Q} \in \mathbb{R}^{(m-2)\times(m-2)}$ raccoglie le probabilità di transizione tra stati non assorbenti, mentre $\mathbf{R} \in \mathbb{R}^{(m-2)\times 2}$ contiene le probabilità di passare da stati non assorbenti ai 2 stati assorbenti e $\mathbf{0} \in \mathbb{R}^{2\times(m-2)}$ è la matrice nulla.
Data una MC descritta dalla matrice $\mathbf{P}$, la matrice $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$ è chiamata *matrice fondamentale* per $\mathbf{P}$. L'ingresso n_{ij} di $\mathbf{N}$ descrive il valore atteso di volte che la catena passerà allo stato s_j partendo dallo stato s_i.
Sia inoltre b_{ij} la probabilità di passare da uno strato i transiente ad uno stato j assorbente, e sia $\mathbf{B}$ (nel nostro caso $\mathbf{B} \in \mathbb{R}^{(m-2)\times 2}$) la matrice composta dalle entrate b_{ij}. Si può provare che:

$$\mathbf{B} = \mathbf{N} \cdot \mathbf{R}$$

Si definiscano i seguenti eventi:

- $\mathcal{A}_{T_m} = \{\text{La random walk parte dallo stato } m \text{ del Tangle}\}$
- $\mathcal{B}_{T_m} = \{\text{La random walk viene assorbita dallo stato } m \text{ del Tangle}\}$
- $\mathcal{A}_{P_l} = \{\text{La random walk parte dallo stato } l \text{ della PC}\}$
- $\mathcal{B}_{P_l} = \{\text{La random walk viene assorbita dallo stato } l \text{ della PC}\}$

Allora vale che:

$$\mathbb{P}(\mathcal{A}_{\mathrm{T}_1} \cap \mathcal{A}_{\mathrm{T}_{\mathrm{n}_{\mathrm{tangle}}}}) = b_{11}$$
$$\mathbb{P}(\mathcal{A}_{\mathrm{T}_1} \cap \mathcal{B}_{\mathrm{T}_{\mathrm{n}_{\mathrm{tangle}}+\mathrm{n}_{\mathrm{parasite}}}}) = \quad b_{12}$$

Ricordando che $n_{tangle} + n_{parasite}$ identifica l'indice dell'ultimo stato della PC, b_{12} è la probabilità che, data una random walk che parte dal Tangle, essa finisca in uno dei tip dalla Parasite Chain. In altre parole, è la probabilità di successo dell'attacco Parasite Chain.
I risultati ottenuti per via analitica non sono del tutto precisi a causa di certe approssimazioni fatte (ad esempio quelle riguardanti il rate di crescita del peso del Tangle o il fatto che una transazione sia approvata da altre due, in questo modello), tuttavia si avvicinano molto ai risultati ottenuti per via numerica.

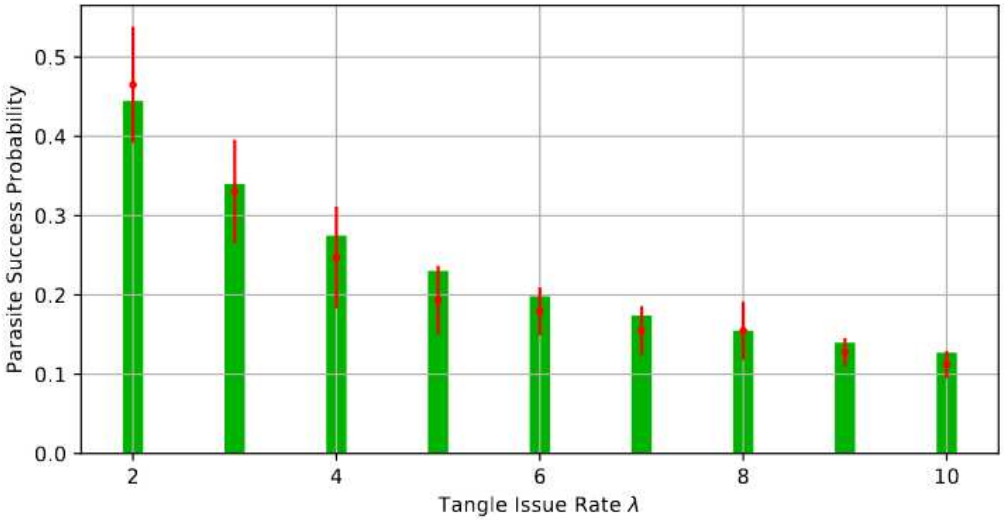


Figura 3.2: In verde i risultati analici, in rosso i risultati numerici. Per questa simulazione i parametri utilizzati sono stati $\lambda = 2, 3, 4, 5, 6, 7, 8, 9, 10$, $\mu = 0.1$, $h = 1$, $q = 0$, $T_1 - T_0 = 80$ e $T_2 - T_1 = 10$.

Da questo grafico, inoltre, si intuisce immediatamente che la probabilità di successo dell'attacco diminuisce all'aumentare di λ. Questo risultato non sorprende, visto che se λ aumenta significa che l'attaccante ha bisogno di maggior potenza computazionale per portare a termine l'attacco.

3.3 Miglioramenti dell'algoritmo

3.3.1 Il termine di bias

Nelle sezioni precedenti è stato menzionato più volte il termine di bias α che compare in

$$P_{xy} = \frac{f(-\alpha(\mathcal{H}_x - \mathcal{H}_y))}{\sum_{z:z\to x} f(-\alpha(\mathcal{H}_x - \mathcal{H}_y))}$$

Poiché è un parametro che deve essere scelto, è di notevole importanza analizzare come cambi la probabilità di transizione P_{xy} al variare di α.
Si possono distinguere innanzitutto i due casi limite:

- $\alpha = 0$: il contributo del peso comulativo delle transazioni alla probabilità P_{xy} è nullo. In altre parole, la probabilità è uniforme, pertanto l'algoritmo MCMC in questo caso equivale all'algoritmo URTS.

- α "grande": facendo attenzione al fatto che $\mathcal{H}_x - \mathcal{H}_y < 0$, si osserva immediatamente che l'algoritmo sceglierà *sempre* le transazioni con peso maggiore, trasformando il Tangle in una catena.

Naturalmente nessuna delle due situazioni è accettabile, visto che da un lato verrebbero annullati tutti i vantaggi dell'algoritmo MCMC e dall'altro si ridurrebbe il Tangle ad una semplice Blockchain. Si consideri ad esempio la situazione in cui i nuovi tip vengano scelti facendo svolgere all'algoritmo una camminata deterministica: la strada che prenderà sarà sempre la stessa, e questo porterebbe, nel tempo, a trasformare il grafo in una catena, visto che non si considererebbero altre strade se non quella che tocca le transazioni con il peso maggiore.
Pertanto è necessario porre l'interesse su quale sia il miglior valore di α, inteso come miglior compromesso tra sicurezza ed efficienza.

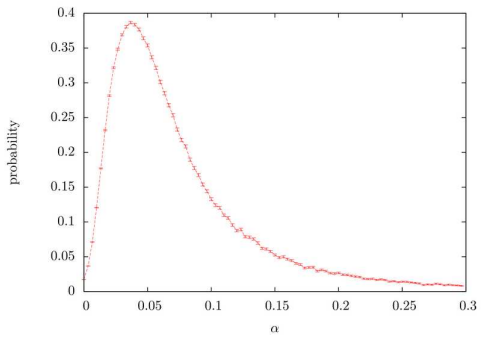


Figura 3.3: Questa simulazione numerica mette in relazione le probabilità di successo del grafico con in possibili valori di α. In questo caso l'attaccante aveva a disposizione il 20% della potenza del sistema.

I risultati del grafico sono evidenti e in una certa misura inaspettati. Per $\alpha = 0$ la probabilità di successo *non è nulla*, ma questo è coerente col fatto che ci si trova a tutti gli effetti con un algoritmo che sceglie i tip con probabilità uniforme. Quello che sorprende di più è certamente il massimo per $\alpha = 0.05$: in base alle osservazioni fatte in precedenza era lecito aspettarsi una funzione monotona decrescente. Questo picco è dovuto al fatto che il successo dell'attacco sia legato al verificarsi di due eventi:

- la random walk deve passare per la radice r
- da r deve scegliere la strada della PC

La probabilità che si verifichi il secondo evento naturalmente diminuisce all'aumentare di α: se l'attaccante non si trova in vantaggio sul sistema, a livello di potenza computazionale disponibile, la PC avrà sempre un peso *inferiore* rispetto al Tangle principale, e quindi, se aumenta α aumenta anche la probabilità che la random walk rimanga nel Tangle principale, perché ha più peso. Tuttavia, per quanto riguarda il primo evento, il fatto che ad r siano collegati diversi vertici della PC contribuisce ad aumentarne il peso, e quindi a rendere più probabile il passaggio della random walk proprio per la radice.
Da quest'analisi è emerso quindi che ci sia un valore di α da evitare assolutamente, anche perché porta con sé una probabilità di successo dell'attacco del 40%, di certo non trascurabile. In generale, la scelta di α può essere fatta cercando il miglior compromesso tra sicurezza del Tangle ed efficienza.

3.3.2 Algoritmo ibrido

Come detto in precedenza, qualora si scegliesse α grande, ci sarebbe ristretto numero di transazioni che vengono approvate, fino ad arrivare al caso limite della catena.
In particolare, questa situazione penalizza pesantemente i tip "vecchi", ovvero quelli che sono stati immessi nel Tangle da tempo ma che non sono ancora stati approvati. I nuovi tip, invece, legandosi a transazioni più recenti, e dunque con più peso, hanno molta più probabilità di essere scelti, rendendo *orfani* gli altri.
Questa problematica tocca indirettamente la sicurezza del Tangle, ma è un argomento di notevole interesse implementativo, visto che non è certamente auspicabile una situazione in cui ci siano numerosi *orfani*, e pertanto una scelta votata a diminuirne il numero, come scegliere α più piccolo, potrebbe intaccare la sicurezza generale del Tangle stesso.

Una possibile soluzione potrebbe essere un algoritmo *ibrido*, che porti con sé i vantaggi di α piccolo e di α grande. Idealmente, si possono individuare due passi fondamentali:

1. Si esegue una selezione di tip con MCMC e α grande, in maniera tale da scartare i tip "sospetti" per prevenire un eventuale double spending.

2. Si esegue una seconda selezione, questa volta con algoritmo URTS o MCMC con α piccolo, per evitare di avere un elevato numero di tip orfani.

Questa scelta implementativa garantirebbe la possibilità di evitare le ripetizioni dell'algoritmo dovute al metodo Monte Carlo. Infatti, con la prima scansione dei tip effettuata dall'algoritmo MCMC con α grande, si scartano tutte le transazioni con poco peso associato e dietro alle quali potrebbe nascondersi una catena di Markov. Con il secondo passo, ovvero la selezione con probabilità uniforme, si evita il quasi-determinismo che deriva dalla scelta di un termine di bias grande, risolvendo così il problema dei tip orfani.
Tuttavia, la debolezza di tale algoritmo ibrido risiede nel primo step: un attaccante, infatti, disponendo di molte risorse computazionali, potrebbe creare una catena di grandi dimensioni, e quindi aumentare notevolmente il peso della radice a cui è collegata nel Tangle principale: è evidente che un attacco PC, sotto queste condizioni, risulterebbe ancora una seria minaccia.

3.3.3 MCMC di "prim'ordine"

Si ricordi che λ e μ sono stati definiti come i rate d'immissione delle transazioni nel Tangle e nella PC, rispettivamente. Per quanto riguarda il peso cumulativo $\mathcal{H}(t)$ di una generica transazione x, cresce linearmente con pendenza λ se tale transazione appartiene al Tangle, mentre ha pendenza μ se appartiene alla catena.

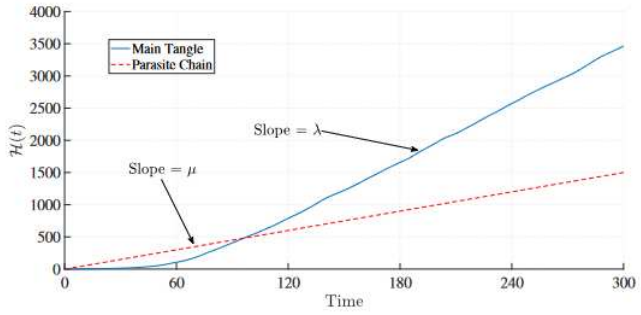


Figura 3.4: Rappresentazione della crescita del peso cumulativo $\mathcal{H}(t)$ nel Tangle e nella PC in funzione del tempo.

Assumendo che $\lambda > \mu$ (altrimenti l'attaccante avrebbe maggior potenza e dunque vincerebbe sicuramente, almeno asintoticamente, come già dimostrato), se si introducono in $P_{xy}^{(t)}$ anche le *derivate prime rispetto al tempo* (da qui MCMC di "prim'ordine") dei pesi cumulativi $\mathcal{H}(t)$, è evidente che a trarne vantaggio sarebbe il Tangle, proprio in virtù del fatto che $\lambda > \mu$. In particolare, si definiscano:

- $\mathscr{H}(t) = |\mathcal{H}_x^{(t)} - \mathcal{H}_y^{(t)}|$
- $\mathscr{H}'(t) = |\mathcal{H}_x^{'(t)} - \mathcal{H}_y^{'(t)}|$ con $\mathcal{H}_x^{'(t)} = \frac{d}{dt}\mathcal{H}_x^{(t)}$ e $\mathcal{H}_y^{'(t)} = \frac{d}{dt}\mathcal{H}_y^{(t)}$
- β un termine di bias relativo ad $\mathscr{H}'(t)$ simile ad α

Allora la probabilità di transizione adottata in questa versione migliorata dell'algoritmo diventa:

$$P_{xy}(t) = \begin{cases} \frac{q}{|\mathcal{A}(x)|} & \text{se } y \in \mathcal{A}(x) \\ (1-q)\frac{f(-\alpha\cdot\mathscr{H}(t-h)-\beta\cdot\mathscr{H}'(t-h))}{\sum_{z:z\in\mathcal{A}(z)} f(-\alpha\cdot\mathscr{H}(t-h)-\beta\cdot\mathscr{H}'(t-h))} & \text{se } x \in \mathcal{A}(y) \\ 0 & \text{altrimenti} \end{cases}$$

Una tale modifica all'algoritmo comporta che α possa essere modulato (e in particolare diminuito) affinché il problema dei tip orfani possa essere risolto, senza che la sicurezza venga ad essere intaccata.

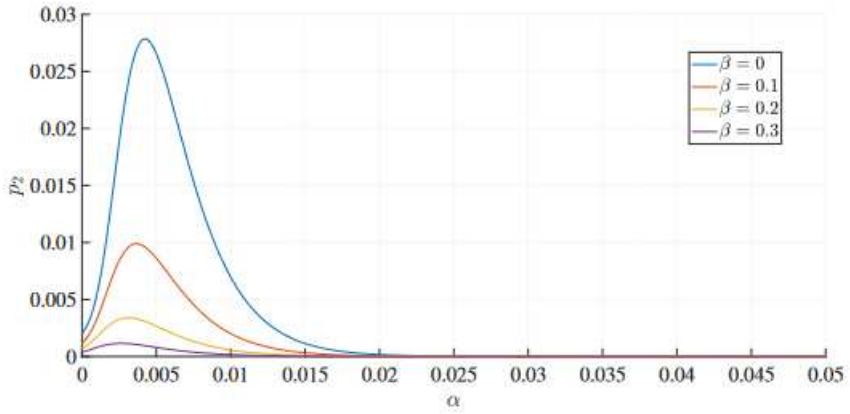


Figura 3.5: Probabilità di successo dell'attacco rispetto ai valori di α per diversi valori di β.

Queste intuizioni sono confermate dalle simulazioni numeriche, che evidenziano effettivamente che dando maggior importanza alle derivate, ovvero per valori di

β grandi la probabilità di successo dell'attacco diminuisce sensibilmente, contribuendo in questo modo a rilassare le condizioni su α per contribuire all'efficienza dell'intero sistema.

Capitolo 4

L'attacco PC in IOTA

Tangle è nato, come struttura dati, per supportare e memorizzare le transazioni di una criptovaluta attiva nel mondo dell'industria dell'IoT, IOTA. Nei primi tre capitoli la trattazione di Tangle si è concentrata quasi esclusivamente su una versione "teorica", quella che era stata concepita con la nascita di IOTA 1.0. Pertanto la prima versione della criptovaluta presentava le peculiarità che sono state descritte sino ad ora (comprese le debolezze agli attacchi PC e le relative contromisure). Tuttavia, poiché IOTA non è un progetto di ricerca ma una realtà industriale, si è evoluto modificando le caratteristiche di Tangle in favore di una maggiore sicurezza e efficienza.

4.1 L'era del Coordinator

A livello commerciale non era possibile contare sulle, seppur ottime, contromisure teoriche, pertanto la soluzione designata è stata quella di inserire un'entità nel sistema: il Coordinator.

4.1.1 La centralizzazione e il consenso White Flag

Il ruolo di questo nodo privilegiato è quello di immettere periodicamente nel Tangle transazioni speciali (chiamate *milestones*) che hanno la funzionalità di garantire la sicurezza dell'intero sistema: ogni transazione approvata, direttamente o indirettamente, da una *milestone* è considerata sicura. Sebbene i compiti del Coordinato non lo rendano eccessivamente invasivo, la sua presenza può essere un problema, visto che rende il Tangle un sistema *centralizzato*, denaturando l'i-

dea che sta alla base dei sistemi a registri distribuiti.
Un'ulteriore conseguenza è il fatto che il meccanismo di consenso debba essere modificato. In precedenza si faceva riferimento al metodo Monte Carlo, per assegnare un indice di affidabilità ad ognuna delle transazioni, basato, a livello probabilistico, sulla ripetizione dell'algoritmo di selezione dei tip. In altre parole, il consenso probabilistico è stato sostituito da un consenso deterministico che pone le proprie garanzie sul fatto che una certa transazione sia collegata o meno ad una *milestone*, diminuendo in questo caso i tempi computazionali ma dando un'impronta significativamente **centralizzata** all'intero sistema: l'interazione non avviene più in maniera diretta tra due utenti ma deve necessariamente passare attraverso un'entità del sistema.

4.1.2 Il passaggio ad UTXO

La sigla "UTXO" sta per "Unspent Transaction Output" e descrive la procedura per cui, oltre a controllare l'ammontare dei fondi di ogni singolo utente, ne monitora anche i movimenti, tenendo traccia della provenienza e della destinazione di ciascuna transazione. Nelle prime versioni di IOTA il sistema conosceva solamente il saldo di ciascun nodo e per questo motivo rilevare tentativi di double spending rappresentava un'impresa difficoltosa. Si supponga lo scenario in cui un certo utente immetta numerose transazioni nel Tangle: se è disponibile solo il dato di quante transazioni abbia immesso, ma non si sa verso quale altro nodo, potrebbe nascere il dubbio che tale utente stia provando a costruire una Parasite Chain, ma quest'ipotesi non è fondata da alcuna prova, visto che il nodo potrebbe aver compiuto un alto numero di transazioni in maniera legittima senza eventuali secondi fini.
Se invece si monitorano anche i movimenti dei fondi di ciascun utente, è molto più facile identificare comportamenti sospetti che potrebbero culminare con un attacco PC.

4.1.3 Relazione con l'attacco Parasite Chain

Il punto di forza dell'attacco PC risiedeva completamente nel fatto che il Tangle fosse un sistema decentralizzato. Infatti, proprio in virtù della decentralizzazione, nel momento in cui gli utenti si trovavano di fronte ad una biforcazione e dovevano decidere, attraverso un meccanismo di consenso, quale dei due sotto-Tangle accettare e quale rifiutare, la scelta era fatta in modo probabilistico, e quindi un

eventuale attaccante aveva la possibilità di fare in modo che fosse preferita la catena parassita al Tangle principale, aumentandone il peso cumulativo e quindi il numero di transazioni all'interno, secondo la propria disponibilità di potere computazionale.
Con l'inserimento del Coordinator, non ci sono più possibilità per l'attaccante di rendere orfano il Tangle principale in favore della propria catena. Supponendo che l'attaccante inizi a costruire la PC collegandosi direttamente ad una *milestone*, tale catena verrebbe resa orfana con l'inserimento di una successiva *milestone* nel Tangle principale.
Inoltre, data la capacità del sistema di tenere traccia anche dei movimenti delle transazioni attraverso il protocollo UTXO, il rischio di incorrere in un attacco PC è notevolmente diminuito rispetto alla prima versione.

4.2 Il Coordicide

Per IOTA, l'inserimento del Coordinator rappresenta una soluzione temporanea per garantire un elevato standard di sicurezza durante le prime fasi di sviluppo del sistema. Per questo motivo i nuovi sviluppi di IOTA prevedono proprio l'eliminazione del Coordinator, da qui il nome della versione 2.0: Coordicide ("assassinio del Coordinator"). Le ragioni per cui si cerca una strada alternativa ad un'entità centrale sono molteplici:

- se il Coordinator cede, tutto il sistema è compromesso (single point of failure).
- chi gestisce il Coordinator (in questo caso la IOTA Foundation) potrebbe avere la possibilità di scegliere di dare una priorità di conferma a certi messaggi piuttosto che ad altri.
- c'è il rischio che un eventuale attaccante simuli un secondo Coordinator immettendo delle false *milestone*.

Naturalmente un cambiamento così radicale comporterà degli effetti a tutti i livelli del Tangle, dal controllo del rate ai problemi di scalabilità. Una delle novità principali è rappresentata dalla separazione del meccanismo di consenso dalla scelta dei tip, che fino ad ora erano accorpati (prima dell'introduzione del Coordinator il consenso si basava sul numero di volte che una tip veniva scelto dall'algoritmo).

4.2.1 Un nuovo meccanismo di consenso

Una delle maggiori problematiche presentate dal meccanismo originario del consenso era legata all'efficienza: era troppo oneroso, in termini di tempo, eseguire molteplici volte l'algoritmo di selezione dei tip per arrivare, attraverso il metodo Monte Carlo, ad assegnare una percentuale di affidabilità ad ogni transazione.
Seguendo un approccio completamente diverso, il nuovo consenso si fonda sull' "opinione" rispetto ad una transazione di una piccola parte di nodi, invece che di tutti gli utenti del network. Per costruire un consenso generale ogni nodo trasmette ai nodi confinanti le proprie opinioni in merito alle transazioni, contribuendo quindi alla diffusione totale di tali informazioni.
Con "opinione", in questa nuova implementazione, si intende una scelta binaria riguardo ad una transazione. In particolare, data una transazione x immessa nel Tangle da un certo utente, se in un certo intervallo di tempo successivo non sono immesse transazioni da parte di quel nodo, allora i nodi danno un consenso **positivo** a quella transazione e a tutte le transazioni da essa approvate. Al contrario, se si registra un eccesso di transazioni proveniente da un certo nodo, gli altri nodi daranno un consenso **negativo** a tale transazione e a tutte quelle che la approveranno. Intuitivamente, attraverso tale procedura l'attacco Parasite Chain è totalmente inefficace: un eventuale attaccante non riuscirebbe a costruire una PC: verrebbe immediatamente bloccato dagli altri nodi.
A livello di risorse, la maggiore preoccupazione quando si tratta di consenso deriva dal tempo necessario affinché i nodi comunichino tra di loro, per questo per IOTA 2.0 sono stati individuati due procedure di votazione il cui obiettivo è quello di ridurre al minimo le tempistiche per raggiungere il consenso.

4.2.2 FPC

Il primo protocollo di votazione proposto si chiama Fast Probabilistic Consensus (FPC). Come detto in precedenza, dato un sistema che ha m nodi, è necessario che si effettui una votazione per prendere una decisione su un valore binario (nel caso di IOTA 2.0, tale valore rappresenta il fatto che una data transazione sia giudicata positivamente o negativamente). Si suppone che ci sia una certa percentuale c di nodi appartenenti all'avversario.
Indicando le due possibilità con i valori 0 e 1, l'obiettivo finale è arrivare ad un consenso da parte di tutti i nodi su uno dei due valori.

Ad intervalli regolari (epoche), ogni nodo interroga k altri nodi ($k << n$) in merito alla loro opinione. Nello specifico, per la prima volta il protocollo seguirà le seguenti fasi:

- in prima battuta, ogni nodo onesto interroga altri j nodi, scelti in modo aleatorio, e registra il numero $\eta_1(j)$ di "1" ricevuti come risposta.
- si introduce la variabile aleatoria uniforme $X_1 \sim \mathcal{U}(a, b)$ con $1/2 \leq a \leq b \leq 1$.
- ogni nodo onesto sceglierà la propria opinione con questo criterio: se $\frac{\eta_1(j)}{k} \geq X_1$ allora selezionerà 1, altrimenti opterà per 0.

Nelle l epoche successive si segue una procedura analoga, sostituendo la variabile aleatoria X_1 con $X_l \sim \mathcal{U}(\beta, 1-\beta)$, $\beta \in (0, \frac{1}{2})$.
Ricordando che si suppone che nel sistema siano presenti dei nodi avversari, si assuma che tali nodi siano a conoscenza delle regole di decisione che i nodi onesti prendono: in virtù di ciò possono adattare il proprio comportamento per trarre i vantaggi che desiderano. Pertanto, l'idea è quella di rendere sconosciute, agli occhi degli attaccanti, le regole di decisione, cambiando di volta in volta e aleatoriamente la soglia di decisione. Un eventuale attaccante, interrogando a sua volta i k nodi, potrebbe riuscire a derivare una proporzione tra "0" e "1", ma non sarebbe in grado di risalire al valore che la variabile aleatoria "soglia" assume. Per questo la soglia X_1 potrebbe essere separata da quella ricavata in precedenza dall'attaccante. Una volta avvenuta questa separazione, i nodi onesti sceglieranno uno dei due valori con alta probabilità, rendendo tuttavia impossibile ricavare, da parte dei nodi disonesti, informazioni sulle effettive scelte.

4.2.3 Cellular Consensus

In questa seconda proposta di sistema di votazioni, ogni nodo è modellato come un automa cellulare. In particolare:

- in caso di transazioni in conflitto, il nodo cambia opinione seguendo l'opinione di maggioranza che i suoi vicini hanno (naturalmente, mentre il protocollo viene eseguito i vicini non cambiano).
- il nodo richiede una "prova" ai vicini, ovvero di condividere l'opinione dei vicini dei vicini, in maniera tale da scoprire se uno dei vicini non è onesto.

- se un nodo è disonesto, ovvero dichiara un'opinione senza averne le prove, verrà eliminato e la sua eliminazione viene comunicata a tutti i nodi del sistema.

Seguendo questa metodologia, si riuscirà ad arrivare ad un consenso generale senza il bisogno di dover raggiungere ogni nodo nel momento della votazione, ma procedendo di volta in volta con piccoli insiemi di nodi.

4.2.4 Selezione dei tip

Nelle versioni precedenti di IOTA, il meccanismo di consenso e la selezione dei tip erano strettamente collegati. Con l'avvento della nuova procedura di consenso, descritta precedentemente con due declinazioni sulla procedura di votazione, è naturale che anche a livello di selezione dei tip ci siano dei cambiamenti.
In particolare, è stato già diffusamente analizzato il ruolo della selezione nei tip nel tentativo di contrastare attacchi che mirassero alla realizzazione del double spending. Tali problematiche, in IOTA 2.0, verranno prese in carico dal nuovo consenso, pertanto la selezione dei tip verrà esonerata dal compito di salvaguardare il sistema da attacchi come l'attacco Parasite Chain. Per questo motivo, è ragionevole abbandonare la selezione dei tip basata sul metodo Monte Carlo, decisamente inefficiente, in favore di un algoritmo molto simile alla versione URTS (Uniform Tip Random Selection) già presentata.
Oltre all'efficienza, con l'algoritmo URTS i nodi sono in un Equilibrio di Nash (visto che ogni nodo sa che gli altri selezionano i tip con probabilità uniforme, non avrebbe motivo di cambiare il proprio comportamento). La problematica maggiore derivante dall'utilizzo di tale algoritmo, dato che in IOTA 2.0 verrà scongiurato il rischio di un attacco PC, è il fatto che non ci sia una penalizzazione per i nodi "pigri", ovvero gli utenti che immettono transazioni nel Tangle ma non collaborano attivamente alla sicurezza del Tangle (per esempio, per immettere una transazione validano una transazioni "vecchie"). Una possibile soluzione potrebbe essere quella di applicare l'algoritmo solo nel sottoinsieme di tip appartenenti a nodi non pigri, ma questa contromisura sarebbe troppo penalizzante nei confronti dei nodi che, per svariate motivazioni, hanno comportamenti "pigri" in rarissime occasioni. Per questo, la nuova versione dell'algoritmo URTS si concentra sul sottoinsieme delle *transazioni approvate* dai tip, e, da quelle, effettua una scelta con probabilità uniforme sul tip da scegliere. In questo modo i nodi pigri

non sono totalmente tagliati fuori, ma hanno al massimo la metà della probabilità di essere scelti rispetto ai nodi non pigri, visto che la loro connessione con una transazione vecchia non viene considerata dall'algoritmo. In altre parole, con questa nuova versione dell'algoritmo di selezione dei tip si promuove il comportamento virtuoso da parte dei nodi, incentivandoli a validare le transazioni più recenti affinché le proprie vengano immesse nel Tangle, ma non si penalizzano drasticamente gli utenti che, per diverse motivazioni, approvano transazioni più vecchie.

Capitolo 5

Conclusioni

In questa tesi è stato introdotto il Tangle, un sistema a registri distribuiti impiegato da una nuova criptovaluta nel mondo dell'IoT, IOTA. Il Tangle è un grafo aciclico diretto, che rappresenta un'evoluzione della classica struttura della Blockchain e ne migliora gli aspetti critici, tra cui scalabilità, tasse e prestazioni. Inoltre, sono stati introdotti gli algoritmi URTS e MCMC, utilizzati dal Tangle per la selezione quali nuove transazioni siano ammesse nel sistema.
In seguito è stato descritto uno dei principali vettori d'attacco a cui può essere soggetto il Tangle, l'attacco Parasite Chain, delineandone i rischi e le caratteristiche. La trattazione si è quindi concentrata sulle principali contromisure adottabili per contrastare tale tipologia d'attacco, comprendendo un meccanismo per la rilevazione basato sulle differenze strutturali tra catena parassita e Tangle principale. Successivamente è stato fornito un modello matematico dell'attacco basato sulle catene di Markov, delineando probabilità di transizione tra gli stati del Tangle e gli stati della catena e definendo la matrice di transizione, da cui è stata ricavata la probabilità di successo dell'attacco. Per quanto riguarda le contromisure, sono stati analizzati i valori ottimi per il termine di bias α che compare nell'algoritmo MCMC, nonché due principali contromisure all'attacco PC: un algoritmo ibrido che coniugasse efficienza ed efficacia, e una versione migliorata dell'algoritmo MCMC.
In ultima istanza sono state descritte le soluzioni commerciali attuate da IOTA per prevenire e contrastare l'attacco PC. In quest'ottica è stato presentato il Coordinator, una soluzione efficiente ai problemi di sicurezza ma che decreta la perdita di decentralizzazione da parte di Tangle, e la nuova versione IOTA 2.0.

Bibliografia

[1] S. Popov, *The Tangle* (2015)

[2] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System* (2008)

[3] N. Benvenuto, M. Zorzi, *Principles of Communication Network and Systems*, John Wiley & Sons Ltd (2011)

[4] A. Penzkofer, B. Kusmierz, A. Capossele, W. Sanders, O. Saa, *Parasite Chain Detection in the IOTA protocol* (2020)

[5] P. Staupe, *Quasi-analitic Parasite Chain absortion probabilities in the Tangle* (2017)

[6] S. Popov, O. Saa, P. Finardi, *Equilibria in the Tangle* (2019)

[7] A. Cullen, P. Ferraro, C. King, R. Shorten, *On the resilience of DAG-based Distributed Ledgers in IoT applications* (2020)

[8] S. Popov, H. Moog, D. Camargo, A. Capossele, V. Dimitrov, A. Gal, A. Greve, B. Kusmierz, S. Mueller, A. Penzkofer, O. Saa, W. Sanders, L. Vigneri, W. Welz, V. Attias, *The Coordicide* (2020)

[9] S. Popov, W. J. Buchanan, *FPC-BI: Fast Probabilistic Consensus within Byzantine Infrastructures* (2020)

[10] P. Ferraro, C. King, R. Shorten, *"IOTA-based Directed Acyclic Graphs without Orphans"* (2020)

[11] *The Coordinator - PoA Consensus* `https://wiki.iota.org/learn/protocols/coordinator/`

[12] *Alpha: playing with randomness* `https://blog.iota.org/alpha-d176d7601f1c/`

[13] *Attack Analysis - The Simple Parasite Chain* https://blog.iota.org/attack-analysis-the-simple-parasite-chain-42a34bfeaf23/