Informatics

Modeling Improvement and Application of DAG-based Blockchain Networks

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Fengyang Guo, MSc
Matrikelnummer 11937900

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dr. Schahram Dustdar

Diese Dissertation haben begutachtet:

George Pallis

Rajiv Ranjan

Wien, 17. März 2025

Fengyang Guo

TU WIEN Informatics

Modeling Improvement and Application of DAG-based Blockchain Networks

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Fengyang Guo, MSc
Registration Number 11937900

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dr. Schahram Dustdar

The dissertation has been reviewed by:

George Pallis

Rajiv Ranjan

Vienna, March 17, 2025

Fengyang Guo

Erklärung zur Verfassung der Arbeit

Fengyang Guo, MSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 17. März 2025

Fengyang Guo

Acknowledgements

I wish to express my profound gratitude to all those who have provided unwavering support throughout the entirety of my doctoral journey.

Foremost, I extend my heartfelt thanks to my esteemed thesis advisor, Univ. Prof. Dr. Schahram Dustdar, for his invaluable guidance, unwavering encouragement, and constructive feedback that have significantly shaped my research and facilitated the realization of my academic goals.

I would also like to convey my deep appreciation to my two co-advisors, Dr. Xun Xiao and Dr. Artur Hecker, for their perceptive comments, insightful suggestions, and substantial contributions to my work. Their diverse perspectives and scholarly acumen have markedly enriched my research, affording me a more comprehensive comprehension of my chosen subject.

My gratitude further extends to the members of my thesis committee, whose generous allocation of time, dedicated effort, and constructive critique have played a pivotal role in enhancing the quality of my research.

I am profoundly thankful to my colleagues and friends, who have provided steadfast support, encouragement, and intellectual inspiration during my arduous thesis endeavor. Their companionship and camaraderie have rendered this academic journey more enjoyable and rewarding.

Finally, I wish to express my deepest gratitude to my family for their unflagging love, steadfast support, and continuous encouragement. Their unwavering belief in me has served as an enduring source of motivation and inspiration, without which the completion of this thesis would not have been possible.

I extend my heartfelt appreciation to all for their pivotal contributions to my academic achievements.

Kurzfassung

Blockchain, eine revolutionäre Distributed-Ledger-Technologie (DLT), hat in den letzten Jahren große Aufmerksamkeit erregt. Durch den Einsatz kryptografischer Techniken gewährleistet die Blockchain die Unveränderbarkeit, Rückverfolgbarkeit, robuste Sicherheit und den Datenschutz von Daten. Sie arbeitet als dezentralisiertes System und schützt so vor Angriffen mit einem einzigen Fehler. Da jedoch das Datenvolumen aufgrund der zunehmenden Verbreitung von IoT-Geräten weiter ansteigt, steht das Blockchain-Ökosystem vor Herausforderungen. Ein Hauptproblem liegt in der Skalierbarkeit. Die Transaktionsverarbeitungskapazität der Blockchain bleibt deutlich hinter der zentralisierter Systeme zurück. Als Reaktion auf diese Herausforderung haben Forscher verschiedene Strategien entwickelt, um die Skalierbarkeit der Blockchain zu verbessern. Ein vielversprechender Ansatz ist die Verwendung einer Datenstruktur in Form eines gerichteten azyklischen Graphen (Directed Acyclic Graph, DAG) im Gegensatz zu der herkömmlichen linearen Kettenstruktur. IOTA, eine bekannte Blockchain, die auf dem DAG-Modell aufbaut, verwendet eine Datenstruktur, die als "Tangle" bekannt ist. Diese innovative Struktur verwendet einen Markov Chain Monte Carlo (MCMC) Random Walk Algorithmus, um neue Transaktionen an das Tangle anzuhängen. Theoretisch sollte ein höheres Volumen an Transaktionen, die dem Tangle zugeordnet werden, zu einer verbesserten Transaktionsrate pro Sekunde (TPS) führen. In der Praxis ergeben sich jedoch mehrere Herausforderungen. Es gibt keinen allgemeingültigen Algorithmus für die Transaktionsauswahl, und das tatsächliche Verhalten und die Struktur des Gewirrs bleiben schwer fassbar. Der MCMC-Algorithmus führt einen einflussreichen Random-Walk-Gewichtungsfaktor ein, der sich auf die Sicherheit und Skalierbarkeit des IOTA-Wirrwarrs auswirkt. Ein größerer Gewichtungsfaktor kann die Sicherheit erhöhen, aber zu mehr unbestätigten Transaktionen führen, während ein kleinerer Faktor unbestätigte Transaktionen reduzieren, aber die Sicherheit beeinträchtigen kann. Das Erreichen einer hohen TPS unter realen Bedingungen erweist sich als ein schwieriges Unterfangen. Während die Blockchain-Technologie ein transformatives Potenzial bietet, stellt die Lösung des Problems der Skalierbarkeit in der Praxis, insbesondere im Kontext von DAG-basierten Systemen wie IOTA, eine komplexe und vielschichtige Herausforderung dar.

In dieser Arbeit führen wir eine umfassende Analyse und dynamische Modellerstellung des realen IOTA-Wirrwarrs durch. Darüber hinaus führen wir eine sichere und skalierbare TSA ein und entwickeln einen leichtgewichtigen Authentifizierungsmechanismus, der in IOTA verwurzelt ist. Unsere übergreifenden Ziele umfassen: (i) die Enthüllung der topologischen Attribute, der Leistungsmetriken und des Generierungsmodells des tatsächlichen IOTA-Tangles, (ii) die Entwicklung eines Algorithmus zur Spitzenauswahl, der auf die einzigartigen Eigenschaften von DAG-basierten Blockchains zugeschnitten ist, und (iii) die Entwicklung einer IoT-Anwendung, die auf dem IOTA-Framework basiert. Unsere Erkundungsreise beginnt mit dem Abruf der authentischen Knäueldatenbank. Wenn wir diese Merkmale mit denen eines simulierten Gewirrs vergleichen, stellen wir erhebliche Unterschiede fest. Darüber hinaus machen wir uns auf die Suche nach größerer Präzision bei der Abbildung der in-degree-Verteilung und der Darstellung der sich entwickelnden Tangle-Topologie innerhalb des IOTA-Bereichs. Die Anwendung verschiedener Long-Tail-Verteilungen zeigt, dass die double Pareto Lognormal (dPLN)-Verteilung ihre Konkurrenten in Bezug auf die Anpassungsgenauigkeit übertrifft. Die in-degree-Verteilung zeigt, dass die Mehrheit der Transaktionen nur eine Genehmigung erhält, was auf eine inhärente Fragilität der Topologie mit einer Fülle von Blowball-Strukturen hinweist. Im folgenden Abschnitt unserer Arbeit konzentrieren wir uns auf die Optimierung der Tip Selection Algorithm (TSA) für DAG-basierte Blockchains. Ein besonderer Schwerpunkt liegt auf der Bewältigung realer Herausforderungen, mit denen IOTA konfrontiert ist. Wir beginnen mit einem schnellen TSA, das auf die Ankunft von Burst-Transaktionen in DAG-basierten Blockchains zugeschnitten ist. Dieser neuartige Ansatz vermeidet den gewichteten Random-Walk-Prozess und berechnet die Wahrscheinlichkeitsverteilung für die Spitzenauswahl im Voraus, was die Auswahlaufgabe erheblich beschleunigt. Anschließend erweitern wir diesen Algorithmus zu einer sicheren und skalierbaren Variante. Nach der Berechnung der Tipp-Auswahlwahrscheinlichkeiten identifizieren und wählen wir abnormale Tipps anhand vordefinierter Schwellenwerte aus und fügen anschließend neue Transaktionen nach dem Zufallsprinzip hinzu. Der von uns vorgeschlagene Algorithmus zur Auswahl von Tipps befasst sich mit zwei kritischen Aspekten: (i) Stärkung des Gewirrs gegen den Einfluss unregelmäßiger Strukturen und (ii) Stabilisierung und Minimierung der Anzahl unbestätigter Transaktionen. Schließlich gipfelt die Arbeit in der Entwicklung eines leichtgewichtigen Authentifizierungsmechanismus auf Basis von Proximity, der für domänenübergreifende IoT-Geräte zugeschnitten ist und auf der IOTA-Plattform basiert. IOTA dient als Repository für die bei der Authentifizierung verwendeten Zertifikate. Wir untermauern die Machbarkeit unseres Vorschlags durch die Implementierung eines kompakten internen Prototypsystems.

Abstract

Blockchain, a revolutionary distributed ledger technology (DLT), has garnered significant attention in recent years. Leveraging cryptographic techniques, blockchain ensures the immutability, traceability robust security, and privacy of data. It operates as a decentralized system, thereby safeguarding against single-point failure attacks. However, as the volume of data, continues to surge due to the proliferation of Internet of Things (IoT) devices, the blockchain ecosystem faces challenges. A key issue lies in scalability. Blockchain's transaction processing capacity significantly lags behind centralized systems. In response to this challenge, researchers have devised various strategies to enhance blockchain scalability. One promising approach involves adopting a Directed Acyclic Graph (DAG) data structure, as opposed to the conventional linear chain structure. IOTA, a renowned blockchain built on the DAG model, employs a data structure known as the "tangle". This innovative structure employs a Markov Chain Monte Carlo (MCMC) random walk algorithm to attach new transactions to the tangle. Theoretically, a higher volume of transactions attached to the tangle should result in improved Transaction per Seconds (TPS). In practice, however, several challenges emerge. There is no universally prescribed transaction selection algorithm, and the true behavior and structure of the tangle remain elusive. The MCMC algorithm introduces an influential random walk weight factor, impacting the security and scalability of the IOTA tangle. A larger weight factor may enhance security but lead to more unconfirmed transactions, while a smaller factor may reduce unconfirmed transactions but compromise security. Achieving high TPS under real-world conditions proves to be a formidable undertaking. As a result, while blockchain technology offers transformative potential, addressing the scalability issue in a practical setting, especially within the context of DAG-based systems like IOTA, presents a complex and multifaceted challenge.

In this thesis, we undertake a comprehensive analysis and dynamic model generation of the real IOTA tangle. Additionally, we introduce a secure, and scalable TSA and devise a lightweight authentication mechanism rooted in IOTA. Our overarching objectives encompass: (i) unveiling the topological attributes, performance metrics, and generation model of the actual IOTA tangle, (ii) crafting a tip selection algorithm tailored to the unique characteristics of DAG-based blockchains, and (iii) designing an IoT application underpinned by the IOTA framework. Our exploratory journey begins with the retrieval of the authentic tangle database. By comparing these characteristics with those of a simulated tangle, we discern substantial differences. Furthermore, we embark on a quest for greater precision in mapping the in-degree distribution and charting the evolving tangle topology within the IOTA realm. The application of various long-tail distributions reveals that the double Pareto Lognormal (dPLN) distribution surpasses its peers in terms of fitting accuracy. The in-degree distribution unveils that the majority of transactions garner just one approval, signifying inherent fragility in the topology with a profusion of blowballing structures. In the subsequent segment of our thesis, we shift our focus towards optimizing the Tip Selection Algorithm (TSA) for DAG-based blockchains. A particular emphasis lies in addressing real-world challenges faced by IOTA. We inaugurate with a swift TSA tailored for burst transaction arrivals in DAG-based blockchains. This novel approach avoids the weighted random walk process and precomputes the tip selection probability distribution, dramatically expediting the selection task. We then extend this algorithm into a secure and scalable variant. After calculating tip selection probabilities, we identify and select abnormal tips based on predefined thresholds and subsequently attach new transactions randomly. Our proposed tip selection algorithm tackles two critical issues: (i) fortifying the tangle against the influence of irregular structures and (ii) stabilizing and minimizing the count of unconfirmed transactions. In the end, the thesis culminates with the design of a lightweight proximity-based authentication mechanism tailored for cross-domain IoT devices, underpinned by the IOTA platform. IOTA serves as the repository for certificates employed during authentication. We substantiate the feasibility of our proposal through the implementation of a compact in-house prototype system.

Contents

Publications

The research presented in this thesis is partly based on the following peer-reviewed publications (i.e., journals and conferences). A full list of publications can be found in Google Scholar Profile[1].

- Fengyang Guo, Xun Xiao, Artur Hecker, and Schahram Dustdar, "Characterizing IOTA tangle with empirical data," in *IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2020, pp. 1–6.
- Xun Xiao, Fengyang Guo, and Artur Hecker, "A lightweight cross-domain proximity-based authentication method for IoT based on IOTA," in *2020 IEEE Global Communications Conference (GLOBECOM) Workshops*, IEEE, 2020, pp. 1–6.
- Xun Xiao, Fengyang Guo, Artur Hecker, and Schahram Dustdar, "Fast Tip Selection for Burst Message Arrivals on A DAG-based Blockchain Processing Node at Edge," in *IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2022, pp. 1373–1378.
- Fengyang Guo, Xun Xiao, Artur Hecker, and Schahram Dustdar, "Modeling Ledger Dynamics in IOTA Blockchain," in *IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2022, pp. 2650–2655.
- Fengyang Guo, Xun Xiao, Artur Hecker, and Schahram Dustdar, "A Theoretical Model Characterizing Tangle Evolution in IOTA Blockchain Network," *IEEE Internet of Things Journal*, vol. 10, no. 2, pp. 1259–1273, 2022.
- Fengyang Guo, Xun Xiao, Artur Hecker, and Schahram Dustdar, "An Efficient Graph-Based IOTA Tangle Generation Algorithm". in *IEEE International Conference on Communications (ICC)*, IEEE, 2023.
- Fengyang Guo, Artur Hecker, and Schahram Dustdar, "A Scalable and Secure Transaction Attachment Algorithm for DAG-based Blockchain," *IEEE Internet of Things Journal, early access, 2024.*

[1] https://scholar.google.com/citations?user=zbAe3scAAAAJ&hl=en

CHAPTER 1

Introduction

Blockchain, a pioneering Distributed Ledger Technology (DLT), has garnered considerable attention in recent years. Initially conceived as a Peer-to-Peer (P2P) transaction system, blockchain distinguished itself by enabling efficient transactions without the need for intermediary entities. Operating within a fully decentralized framework, it effectively mitigated the risk of a single point of failure. In this system, each participant maintains a synchronized copy of the ledger records. The information stored within the blockchain is inherently resistant to tampering and, crucially, is marked by irrevocable immutability. Once data is inscribed on the blockchain, it remains unalterable. Consequently, blockchain's unique attributes have rendered it applicable to a diverse array of domains, including finance, Internet of Things (IoT), and the realization of smart cities. As the volume of data within the blockchain ecosystem continues to escalate, certain limitations have become apparent. A pronounced shortcoming is the challenge of scalability, leading to reduced transaction confirmation speeds. In contrast to centralized systems, which offer swift transaction processing, blockchain systems are notably slower. For instance, VISA boasts a transaction processing rate of 10,000 Transactions per Second (TPS). To address this issue, a spectrum of solutions has been proposed, including sharding, Layer 2 scaling solutions, and the adoption of the Directed Acyclic Graph (DAG) data structure. The incorporation of DAG into the blockchain architecture represents a fundamental transformation of the traditional data structure, offering a promising avenue for enhancing scalability.

The realm of blockchain technology has witnessed a paradigm shift with the advent of DAG-based blockchains, which revolutionize the manner in which transactions are processed. In these innovative systems, transactions are structured as a DAG rather than a linear chain, and they have engendered the development of various DAG-based blockchains, including notable platforms such as IOTA, Byteball, Hashgraph, and Fantom. Among these, IOTA has emerged as one of the most prominent and is stewarded by the IOTA Foundation (IF). IOTA employs a distinct DAG data structure, referred to as the 'tangle', which serves as its foundational ledger and repository for all transactions. Distinguished by its absence of transaction fees and robust support for micro-transactions, IOTA stands out as a scalable solution. Theoretically, as the number of transactions integrated into the tangle increases, the transaction speed accelerates, rendering IOTA particularly well-suited for IoT use cases. Within the tangle, each transaction is represented as a vertex, commonly termed a 'site'. In the present landscape, IOTA exists in two primary iterations: the older IOTA 1.0 and the more recent IOTA 2.0. Despite the evolution to IOTA 2.0, the unique consensus mechanism of IOTA 1.0 continues to captivate significant attention and interest. Several research efforts and applications have been developed around IOTA 1.0, underscoring its enduring relevance. This paper focuses specifically on the examination and exploration of IOTA 1.0. Henceforth, the term 'IOTA' herein refers to 'IOTA 1.0'.

IOTA relies on the Markov Chain Monte Carlo (MCMC) as its consensus mechanism. Within the tangle, a novel transaction undergoes a stochastic selection process. A 'walker' embarks on a random traversal from a predefined site to the extremity of the tangle, culminating in the selection of a 'tip', which denotes an unconfirmed transaction. Various Tip Selection Algorithm (TSA) have been devised, with the MCMC standing as the official and recommended TSA. Subsequent to the initial tip selection, the random walk repeats, leading to the selection of a second tip, whereby the new transaction becomes appended and confirmed through its connection to these chosen tips. Each site within the tangle is endowed with a crucial attribute known as cumulative weight, denoting the number of sites linked to the respective site. A pivotal parameter governing the course of the random walk is α, which influences the direction taken and, concomitantly, the tangle's security. A higher α value compels the random walker to traverse along routes with substantial cumulative weights, ultimately permitting the selection of tips along such paths. This, however, results in an elevated count of unconfirmed transactions. Paradoxically, the tangle's security is bolstered, as transactions associated with greater cumulative weights manifest heightened confidence and broad approver support in comparison to their counterparts with lesser cumulative weights.

1.1 Problem Statement

The utilization of the DAG data structure empowers IOTA to engage in asynchronous transaction processing, allowing multiple transactions to be appended to the blockchain concurrently rather than in a sequential manner. However, the adoption of the DAG data structure introduces greater complexities and poses a heightened demand on the consensus mechanism. The asynchronized attachment of transactions introduces increased diversity, potentially resulting in amplified security concerns. Within this section, we delineate the scope of this thesis by articulating a comprehensive description of the challenges that form the focal point of our research.

The primary objective of this thesis is to develop a consensus mechanism that achieves both security and scalability within the context of DAG-based blockchains. More specifically, our aim is to enhance the efficiency of attaching new transactions, ensuring swift and streamlined processes while concurrently regulating the number of unconfirmed transactions, and maintaining a consistent and manageable level. Our research unfolds in a multifaceted approach, encompassing several critical dimensions of IOTA. We intend to (i) conduct a comprehensive analysis of the genuine IOTA tangle to unveil its inherent characteristics, (ii) explore the degree distribution and generation processes of the authentic IOTA tangle, (iii) introduce a highly efficient and secure transaction attachment algorithm, and (iv) probe the potential utility of IOTA within the telecommunications domain. The proposed algorithm offers the dual benefits of reducing computational complexity during the attachment of new transactions and upholding the overall security and stability of the tangle. Moreover, we will provide a tangle simulator to facilitate the rapid generation of tangles for research purposes, ensuring accessibility for both academic researchers and private enthusiasts seeking to engage in tangle analysis.

IOTA employs a DAG data structure, allowing for parallel transaction attachment to the tangle, thereby enhancing the blockchain's operational efficiency. Previous research endeavors have primarily evaluated IOTA through simulation-based methodologies. For instance, in the work by [KSG18a], an analysis of cumulative weight development, in-degree distribution, and the number of tips was conducted, which included the creation of an offline IOTA network for performance testing. This research substantiated IOTA's commendable scalability and TPS capabilities. However, the practical deployment of IOTA confronts a fundamental challenge in that it lacks a prescribed and universally accepted TSA, leading to diverse TSA implementations and subsequently varying tangle topologies. This variation underscores the criticality of understanding the authentic structure and performance of IOTA in real-world scenarios, which is presently unknown. A comprehensive comprehension of IOTA's genuine performance is an imperative prerequisite for any attempts aimed at enhancing its efficacy.

IOTA utilizes a DAG data structure as its foundational framework. This architectural choice empowers IOTA to execute the concurrent attachment of transactions to the tangle, consequently enhancing the operational efficiency of the blockchain. Prior academic inquiries into the domain of IOTA have predominantly relied upon simulation-based methodologies to assess its performance. For instance, as exemplified in [KSG18a], comprehensive analyses pertaining to the development of cumulative weight within the tangle, in-degree distribution, and the quantification of tips were undertaken. In the mentioned reference, an offline IOTA network was meticulously engineered to facilitate performance evaluations, ultimately confirming IOTA's commendable scalability and TPS capabilities. However, the practical implementation of IOTA confronts a fundamental quandary as the IOTA Foundation refrains from mandating a universally recognized TSA, thereby giving rise to a proliferation of diverse TSA implementations and, in turn, varying tangle topologies. This scenario underscores the critical imperative of comprehending the genuine structure and performance of IOTA within authentic, real-world contexts, a dimension that remains elusive. To elevate IOTA's operational efficiency, a profound understanding of its real-world performance is an essential prerequisite.

After knowing the real IOTA generation process and dynamic model, we could consider improving the performance of the IOTA consensus mechanism. Due to the asynchronized attachment process and the uncertainty of the random walk, it is hard to improve and design the consensus mechanism. The main difficulties are 1) the influence of the random walk factor on the unconfirmed transaction, and 2) the change of the new transaction coming rate, 3) the network delay during the synchronization between different nodes. Hence, the attachment process of the new transaction has much uncertainty and the number of unconfirmed transactions is also influenced. In G-IOTA [BGP19] and E-IOTA [BHP20], DA-IOTA [RID$^+$23] , the original IOTA was optimized. However, these optimizations use a small random walk influence factor α to stabilize the number of unconfirmed transactions. Through a small α, the unconfirmed transactions could be at a stable and low level. However, for a small α, some abnormal transactions could have a bigger probability of being chosen. The whole system is still under a risk of attack. We still need a secure transaction attachment mechanism.

The introduction of a novel transaction attachment mechanism necessitates comprehensive evaluation through simulation and testing within the IOTA framework. Presently, simulation tools are predominantly grounded in the IOTA protocol and rely on a random walk process to generate the tangle. However, these existing simulators require substantial computational resources, consuming considerable energy and time during the process of recalculating the random walk for tangle generation. Therefore, the imperative for a more efficient and resource-conserving simulator becomes evident, aimed at economizing both time and energy resources.

1.2 Research Questions

The challenges delineated above serve as the foundational impetus driving the research conducted throughout the course of this thesis. Within the expanse of this study, we elucidate the pivotal research inquiries that form the bedrock of our investigation and subsequently provide answers to these questions.

Q1: What are the structural attributes and performance characteristics of the authentic IOTA tangle, and what constitutes the dynamic generation model of the real IOTA tangle?

In theory, the DAG-based IOTA tangle exhibits superior scalability when compared to traditional chain-based blockchains. The augmentation of transactions attached to the tangle theoretically leads to heightened confirmation speeds, rendering IOTA a highly promising candidate for IoT applications. However, real-world scenarios deviate from simulated environments, necessitating the acquisition of data from actual IOTA operations to scrutinize the performance and generation processes of the real IOTA tangle. Distinguished by its distinctive DAG-based architecture, IOTA diverges from traditional chain data structures featuring a singular linear chain. Compounded by the absence of an official, universally mandated tip selection algorithm for IOTA, the genuine IOTA tangle manifests distinctions from its simulated counterpart. Consequently, the precise topology and performance of the authentic IOTA tangle, along with the intricacies of its real-world generation process, remain elusive. Gaining insight into the genuine IOTA performance and generation process holds the potential to facilitate optimization of the tangle generation algorithm, ensuring a more seamless alignment of the IOTA tangle with real-world use cases.

Q2: How can an efficient and secure transaction selection algorithm be developed for execution on resource-constrained devices within the context of a DAG-based blockchain?

Challenges persist when it comes to attaching new transactions to the tangle in resource-constrained and asynchronized systems. The primary challenge lies in achieving the delicate balance of maintaining security while simultaneously regulating the quantity of unconfirmed transactions, ensuring it remains at a lower and consistent level. IOTA's utilization of the MCMC random walk in its tip selection algorithm imparts a transaction weight, with the random walk inherently influenced by this weight. This influence is governed by a particular factor, which plays a pivotal role in determining the extent of influence. A larger factor enhances the impact of transaction weight, compelling the random walk to traverse the route where transactions with substantial weight are concentrated. Transactions with significant weight boast more endorsements, thereby enhancing their security but leading to a proliferation of unconfirmed transactions. Conversely, adopting a smaller factor mitigates this bias, infusing greater randomness into the process, but it can compromise security. Consequently, the design of a tip selection algorithm that balances security and the number of unconfirmed transactions becomes imperative for optimizing the system's utility.

Q3: How can an efficient simulator be designed to rapidly generate IOTA tangles for the analysis of TSA performance?

Upon the development of a novel tangle transaction selection algorithm, the need for a simulator arises, enabling the generation of tangles and the subsequent evaluation of the algorithm's performance through the analysis of the generated tangles. Conventionally, simulators generate the tangle by iterating the random walk process. However, the generation of numerous tangles using this approach demands substantial time and energy resources. Therefore, the imperative for an efficient simulator that expedites tangle generation becomes apparent.

Q4: How can IOTA be effectively deployed in real-life IoT use cases and contribute to the enhancement of IoT service performance?

In the evolving landscape of the IoT, the proliferation of IoT devices poses considerable challenges, particularly in terms of inter-machine communication. Among these challenges, cross-domain authentication between machines emerges as a pivotal concern. Conventional centralized authentication methods may fall short in meeting the demands presented by the extensive number and dispersed nature of IoT devices. As previously discussed, IOTA exhibits enhanced scalability and has been purposefully designed to cater to the requirements of IoT. Serving as a fundamental component of the IoT network, it becomes essential for IOTA to furnish a lightweight authentication mechanism, specifically tailored to the unique needs of IoT devices.

1.3 Scientific Contributions

In this section, we delineate the contributions made by this thesis to the state of the art by addressing the previously stated research questions. These contributions encompass:

C1: A comprehensive series of analyses and investigations were conducted on the IOTA tangle.

These analyses and investigations include an in-depth examination of the real IOTA tangle using historical empirical data sourced from the IOTA mainnet, the formulation of a dynamic model delineating the genuine IOTA tangle generation process, and an accurate portrayal of the in-degree distribution within the authentic IOTA tangle.

Due to the diversity of TSA and the variances in node behaviors, the real IOTA tangle exhibits distinctions when compared to its simulated counterpart. Through initial detection efforts, anomalous structural patterns were identified within the real IOTA tangle. Consequently, a comprehensive analysis was undertaken to comprehend the real IOTA tangle's topology and performance, with a particular focus on confirmation delay within authentic IOTA transactions. The tangle was meticulously reconstructed based on real IOTA data, followed by an analysis of in-degree distribution, cumulative weight, and transaction delay. This investigation revealed substantial differences between the real tangle and its simulated counterpart, notably the presence of greater delays than those theoretically expected.

Furthermore, the Stochastic Differential Equation (SDE) model was harnessed to elucidate the genuine tangle generation process. In this endeavor, the in-degree distribution was compared against common long-tail distributions, with the empirical finding that the in-degree distribution in the real tangle adheres to the double Pareto Lognormal (dPLN) distribution. To refine this discovery, the Expectation-Maximization (EM) fitting algorithm was employed to fine-tune the distribution, thereby rendering it a more precise fit. The performance of this fitting was evaluated and contrasted.

Detailed information regarding this contribution is presented in Chapter 3, 4, 5 and was originally introduced in [GXHD20a, GXHD22b, GXHD22a].

C2: An optimization of the transaction selection algorithm: a faster, secure and scalable, supporting the attachment of burst coming transactions algorithm.

In real-world scenarios, nodes often encounter the simultaneous arrival of numerous transactions, all vying for attachment to the tangle. Traditionally, transaction selection algorithms processed these transactions sequentially, necessitating at least two separate random walk operations for each new transaction. This approach led to increased transaction attachment delays. To address this challenge, a fast transaction selection algorithm was devised, specifically tailored to address burst-arrival scenarios. This algorithm eliminates the need for repetitive random walks. When confronted with a batch of new transactions entering the tangle concurrently, the node utilizes a predefined list of transactions with corresponding selection probabilities for efficient decision-making. Moreover, building upon this foundation, a secure and scalable transaction selection algorithm was formulated. This advanced algorithm possesses the ability to identify abnormal transactions promptly and subsequently attaches incoming transactions to the tangle in a random manner. This approach serves to maintain the count of unconfirmed transactions at a low and stable level.

Detailed information regarding this contribution can be found in Chapter 6, 7 and was initially introduced in [XGHD22].

C3: An efficient tangle simulator algorithm generating tangles without repeating random walks.

In this contribution, we have optimized the tangle simulation process, eliminating the requirement for random walks. We introduce the Graph Generation and Refinement algorithm (GraGR). With GraGR, tangles can be generated directly by inputting predefined parameters and applying graph theory principles. This approach streamlines the tangle generation process, rendering it free from random walks and significantly saving both time and energy resources. Comparative analyses between tangles generated using GraGR and those created by traditional tangle generators, which rely on simulating the IOTA tangle principles, reveal similar properties. However, GraGR demonstrates superior efficiency when compared to conventional simulators.

Detailed information regarding this contribution can be found in Chapter 8 and was initially presented in [GXHD23a].

C4: An exploration of IOTA use cases in the field of IoT, particularly the implementation of lightweight machine-to-machine authentication.

IOTA's potential applications in the realm of IoT have been extensively studied, spanning domains such as smart homes, internet of vehicles, and smart factories. Our research specifically delves into the utilization of IOTA within the context of Mobile Edge Computing (MEC). We have integrated IOTA and MEC to establish an authentication network that involves multiple IoT service providers. In a more detailed perspective, we have defined customized transaction content and the core operational procedures. This system empowers authentication at resource-constrained local devices, eliminating the need for centralized servers. Additionally, to validate the viability of our proposed lightweight authentication solution, we have constructed a small in-house prototype.

A comprehensive exploration of this contribution can be found in Chapter 9 and was originally introduced in [XGH20].

1.4 Thesis Structure

This thesis is built upon the contributions of original research papers published in journals and conferences. Certain chapters have been expanded or refined to better align with the overall context of the thesis. The structure of this thesis is as follows:

- Chapter 2 provides a foundation by offering background information and introducing the concepts and terminology that will be referenced throughout the thesis.
- Chapter 3 offers a comprehensive analysis using real transaction data provided by the IOTA Foundation. It highlights the existing gaps in IOTA's ability to meet the stringent requirements of delay-sensitive IoT applications.
- Chapter 4 presents a generative model for the IOTA tangle, employing stochastic analysis.
- Chapter 5 introduces a theoretical model for the evolving IOTA tangle based on stochastic analysis. After analyzing real-world IOTA snapshots, a key finding emerges: the IOTA tangle follows a dpln degree distribution. Furthermore, we estimate model parameters through a newly designed fitting algorithm based on the EM algorithm.
- Chapter 6 unveils a novel tip selection algorithm tailored for the scenario of burst message arrivals on edge nodes.
- Chapter 7 introduces a scalable and secure transaction attachment algorithm designed for DAG-based blockchains.
- Chapter 8 presents a lightweight cross-domain authentication mechanism for IoT, built upon IOTA.
- Chapter 9 brings the thesis to a conclusion, summarizing our contributions and offering insights into potential future work.

CHAPTER 2

Preliminary

In this chapter, we present an overview of the fundamental concepts that underlie our work, encompassing IOTA-related concepts, including IOTA tangle, TSA, and common attacks separately. Additionally, we provide a comparative analysis of scalability solutions.

2.1 IOTA Tangle

The IOTA tangle is a ledger of IOTA that comprises transactions and directed links connecting these transactions. The directed link between two transactions signifies an approval relation and also denotes the order of attachment. The more transactions that attach to a particular transaction, the greater the confidence that transaction acquires. The transaction that lacks any referred transactions is deemed unapproved and is referred to as tips. The key idea behind IOTA is that a new transaction validates two previous transactions. As a result of this, linked transactions are disseminated throughout the entire network, leading to the convergence of tangles and the formation of consensus opinions through a distributed consensus protocol.

In the tangle, each transaction possesses its own weight and a concept known as Cumulative Weight (CW). The own weight is assigned a value of 1, while the CW is determined by the number of children of the transaction plus itself. The CW value serves as an indicator of a transaction's significance within the tangle. A higher CW value implies that the transaction has received more approvals compared to transactions with lower CW values. The difference between the CW values of two connected transactions is referred to as the Edge Weight (EW).

2.2 Transaction Attachment

A new transaction (cf. the white square in Figure 2.1) is composed at a client and submitted to a node. An initial validation is done by checking some attributes that can be locally verified such as signature, balances and so on. A validated transaction is attached to two *tips* selected from the tangle, where a tip is a site that is not yet referenced (i.e., approved) by any other site (e.g. Tip m_3, m_4, m_5 in Figure 2.1).

IOTA attaches new incoming transactions to the tips through the TSA, which is executed by a node locally. The official recommended TSA is the MCMC algorithm, which selects tips through a biased random walk process. A random walker initiates its walk from a predefined beginning transaction towards the end of the tangle, i.e., the tip. An important parameter in the MCMC algorithm is α, which influences the probability of tip selection. A large α value causes the random walk to prioritize tips with high cumulative weight, resulting in more unconfirmed transactions. Conversely, a small α value leads to a more random walk process. An α value of 0 results in an unbiased MCMC. Another commonly used TSA is the Uniform Random Tip Selection (URTS) algorithm, which selects tips randomly from the tip pool. Once a new transaction attaches to the tips, this new transaction becomes a new tip and the selected tips are approved and no longer available for selection. While there is no mandatory TSA, IOTA Foundation recommends the use of MCMC for better security and stability of the tangle. URTS and Unbiased Random Walk (URW) are theoretical TSA and cannot be used in the real-life implementation of DAG based DLT due to their vulnerability to parasite chain attacks [KSP+19a].

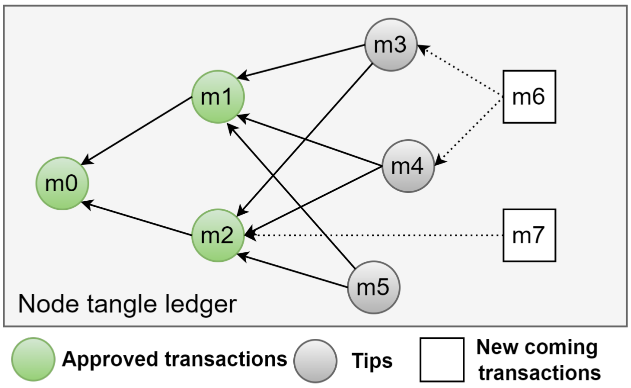


Figure 2.1: Transaction attachment on the tangle

Here we provide a detailed illustration of MCMC, as shown in Figure 2.1, m_3, m_4, and m_5 represent tips, while m_6 and m_7 denote new incoming transactions. A random walker walks from m_0 towards the end of the tangle. The transition probability between m_0 and m_1 is calculated using Equation 2.1. By following the same approach, we can calculate the probability of other edge transactions. Finally, m_3 and m_4 are selected by m_6 via MCMC.

$$p_{m_0 m_1} = \frac{e^{-\alpha EW_{m_0 m_1}}}{e^{-\alpha EW_{m_0 m_1}} + e^{-\alpha EW_{m_0 m_2}}} \tag{2.1}$$

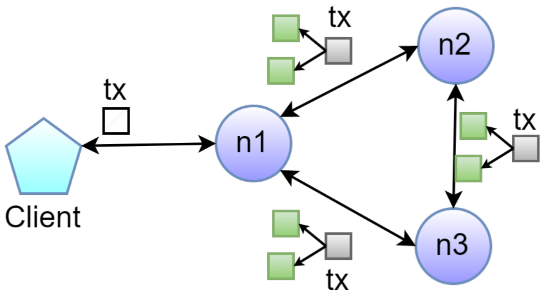


Figure 2.2: Transaction Propagated to Other Nodes

2.2.1 IOTA Transaction Propagation

In parallel, a node receives forwarded transactions from neighboring nodes (as shown in Figure 2.2). A forwarded transaction could already exist in the local ledger. In this case, the node ignores it locally, but forwards it to all neighbors, except to the expedient. If the transaction does not exist in its local ledger (e.g., node n_2 does not have the transaction received from node n_1), a node (here: node n_2) saves the transaction and checks, whether the two referenced sites (denoted as two small eclipses on the transaction) can be found in its tangle. If so, the node simply adds the transaction to its tangle; otherwise, the transaction is suspended, until the missing sites are provided from neighbors.

The node will send requests to neighbors to find a missing site. For example, if node n_2 does not have the two sites, it will broadcast requests to both nodes n_1 and n_3. If any node knows any of the requested transactions, it (e.g., node n_1) it replies to the requester (i.e., node B). Note that this could recursively trigger further missing site requests. Hence, missing sites in a local tangle are progressively completed with the helps of other nodes, until tangles are synchronized.

2.3 Attacks in IOTA

2.3.1 Lazy tip

The lazy tip is a new coming transaction that approves previously approved transactions instead of unapproved ones. While the lazy tip does not contribute to the confirmation rate and does not aid the IOTA system, it does occupy storage space and interaction bandwidth. For instance, in Figure 2.1, transaction m_7 would be identified as a lazy tip, as it approves the already approved transaction m_2.

2.3.2 Parasite chain

An attacker secretly constructs a sub-tangle that cites a transaction on the main tangle, thereby enhancing the cumulative weight of that transaction, as depicted in Figure 2.3.

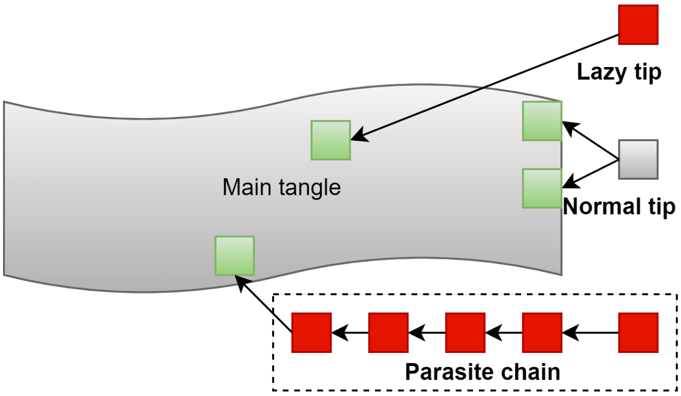


Figure 2.3: Lazy tip and parasite chain

This parasite chain exerts influence on the MCMC random walk process, directing the walker towards the tips on the parasite chain. Consequently, incoming transactions will validate the tips on the parasite chain, while disregarding those from honest nodes. In the worst-case scenario, the parasite chain may reference a double-spending transaction, thereby attracting additional transactions to validate it, ultimately resulting in an attack on the tangle.

2.4 Comparison of Scalability Solutions

Scalability is a critical challenge in blockchain networks, as increasing transaction volumes often lead to congestion, high fees, and reduced efficiency. Several solutions have been proposed to address this issue, including DAG, Sharding, and Layer-2 solutions. This section provides a comparative analysis of these approaches, evaluating their strengths, weaknesses, and applicability to different blockchain ecosystems. Table 2.1 provides a structured comparison of these solutions, highlighting their key differences and use cases.

Criterion	**DAG**	**Sharding**	**Optimistic Rollups**	**ZK-Rollups**
Throughput	High	High	High	High
Decentralization	Varies	Relatively High	Inherits Layer-1	Inherits Layer-1
Security	Requires additional security measures	Cross-shard security required	Fraud proofs required	Cryptographic security
Complexity	High	High	Medium	High
Finality	Probabilistic	Consensus-dependent	7-day challenge period	Immediate
Use Cases	IoT, micropayments	Smart contract platforms	DeFi, general transaction scaling	High-security applications

Table 2.1: Comparative Analysis of Scalability Solutions

2.4.1 Directed Acyclic Graph

DAG-based architectures depart from traditional linear blockchain structures by organizing transactions in a graph-like format. In DAG systems, transactions confirm multiple prior transactions rather than being sequentially added in blocks. Notable DAG-based projects include Avalanche, Nano, and IOTA. They have the following characteristics:

- **Throughput:** High, as transactions can be processed in parallel.
- **Decentralization:** Varies by implementation, consensus mechanisms such as Avalanche's voting or IOTA's coordinator affect decentralization.
- **Security:** Requires additional mechanisms to prevent attacks, such as double-spending.
- **Complexity:** More complex than traditional blockchains, making standardization challenging.
- **Finality:** Non-deterministic, often relying on probabilistic confirmation models.
- **Use Cases:** Well-suited for high-throughput environments, such as IoT networks and micropayments.

2.4.2 Sharding

Sharding enhances scalability by partitioning the blockchain network into multiple parallel chains (shards), each processing a subset of transactions. Ethereum 2.0 and Polkadot utilize sharding as a fundamental scalability strategy. They have the following characteristics:

- **Throughput:** High, as multiple shards operate in parallel.
- **Decentralization:** Maintained, but inter-shard communication can introduce bottlenecks and centralization risks.
- **Security:** Relies on robust cross-shard communication mechanisms, such as Ethereum 2.0's beacon chain.
- **Complexity:** High, due to challenges in shard communication and data consistency.
- **Finality:** Depends on the consensus mechanism; Ethereum 2.0 employs Casper FFG for finality.
- **Use Cases:** Ideal for large-scale smart contract platforms such as Ethereum 2.0 and Polkadot's heterogeneous chains.

2.4.3 Layer-2

Layer-2 scaling solutions operate on top of the main blockchain (Layer-1), reducing on-chain transaction load while leveraging the security of the underlying blockchain. One of the most famous Layer-2 solutions is the Rollups, which aggregate multiple transactions into a single batch, posting compressed data onto Layer-1. Two major types exist: Optimistic Rollups (Arbitrum, Optimism) and ZK-Rollups (zkSync, StarkNet). They have the following characteristics:

- **Throughput:** High, though constrained by Layer-1 data availability.
- **Decentralization:** Inherits Layer-1 security, but centralized sequencers can be a concern.
- **Security:** Optimistic Rollups rely on fraud proofs, whereas ZK-Rollups use cryptographic proofs for enhanced security.
- **Complexity:** Optimistic Rollups are simpler to implement, while ZK-Rollups require advanced cryptographic computation.
- **Finality:** Optimistic Rollups have delayed finality due to challenge periods (typically 7 days); ZK-Rollups achieve near-instant finality.
- **Use Cases:** Suitable for DeFi applications (Optimistic Rollups) and high-security, high-performance environments (ZK-Rollups).

In summary, the key idea behind IOTA is that new transactions approve existing transactions, which progressively gain on weight as more sites approve them. Through transaction propagation, tangles on different nodes mix, grow and finally converge to one. Invalid transactions are blocked locally. The reason for this is that attaching invalid transactions is futile, even for malicious nodes, as during propagation, these will not pass the validation phase on any (honest) node. Recent theoretical analysis has proved the convergence and equilibrium of the tangle synchronization in IOTA [PSF19a].

CHAPTER 3

Characterizing IOTA Tangle with Empirical Data

IOTA organizes transactions in the ledger as a DAG called Tangle, instead of a hash chain of transaction blocks used by most of traditional blockchains. IOTA is considered a promising platform to support IoT applications with its key features such as micro-payment support and absence of transaction fees. While prior art shows extensive analysis based on synthetic data generated through simulations, an analysis based on empirical data from a deployed IOTA network is still missing. In this chapter, we provide the first comprehensive analysis by using real transaction data officially published by IOTA Foundation. Our key finding is that neither the tangle's topological features nor the actual observed performance is consistent with the main conclusions from the literature. In particular, most of transactions take roughly 10 minutes to be officially confirmed, which is not exactly instant as commonly assumed; yet, what is arguably worse is that there is a certain amount (5%) of transactions experiencing exceptionally long confirmation time. This shows that IOTA still has gaps to meet the stringent requirements of IoT applications that are delay sensitive.

The structure of this chapter is outlined as follows. In Section 3.2, we provide a literature review; Section 3.3 introduces our analysis methodology; full results are presented in Section 3.4 and Section 3.5 concludes the chapter.

3.1 Introduction

Blockchain technology enables distributed consensus and is regarded as the ultimate tool to establish a trustworthy relationship in a large-scale anonymous environment. Recently, blockchain technology is being adopted by many industry sectors from finance, logistics, decentralized web services and so on [Pil16].

In 2016, IF[1] proposed a blockchain network, namely IOTA, using DAG (called *tangle*) as the ledger data structure to organize transaction data on every IOTA node [Pop16]. In a tangle, a vertex, namely *site*, represents a single transaction object. A direct edge from one site pointing to another site indicates that the source site approves the destination site. Regarding its consensus mechanism, IOTA removes the Proof-of-Work (PoW) mining phase used in traditional blockchain. Instead, IOTA allows every node to update its local ledger immediately where a new site (i.e., a new transaction) is attached into the tangle by approving two existing sites (called *tips*) in the ledger. Technically, which two tips are selected is not arbitrary but determined by a TSA, wherein the TSA executes two weighted random walks in the tangle until two tips are identified. In IOTA, every IOTA node receives transactions from clients, adds them into its tangle and keeps propagating the processed transactions to its neighbors. As a result, every transaction is propagated across the entire IOTA network, where the distributed tangle ledgers converge to a synchronized status. The mechanism of IOTA will be revisited with more details in the next section.

The key feature of IOTA is its lightweight transaction processing manner without a heavy PoW mining phase. For this reason, IOTA and its variants seem suitable for IoT applications, wherein tiny, massive and ‘instant’ transactions are typical. For example, IOTA is used as a marketplace where electricity trading is directly done by IoT devices as sellers and buyers in [PCAM19]. Another example is IOTA usage in vehicular communication [BVF18a].

Previous studies extensively analyzed IOTA using synthetic data [Kus17, KSG18a, KG18, BRP18, FKCM19a, PSF19a, GRW20]. These works build their own applications and evaluate system performance using the transaction data generated in a simulated environment. However, an analysis based on empirical data generated in the IOTA *mainnet*, i.e., the official IOTA network on the Internet, is still not available in the research community. Consequently, many questions remain open, such as the real tangle topology, the actual transaction confirmation rate in the deployed system, and, in case of diverging findings, the reasons behind the present observations. In this chapter, we try to answer all these questions. More concretely, our main contributions are:

- Since the published transaction datasets do not explicitly contain topology information, we first fully reconstruct all the ledger tangles through identifying all sites and directed edges by looking for every approval relationship among all transactions. In total, 96 tangles were reconstructed from a 322GB original dataset.

- With the reconstructed tangle ledgers, then we compute interested properties based on graph theory and IOTA specification. Specifically, we analyze the diameters, in-degree distribution, cumulative weight of the ledger tangles and measure the actual performance regarding transaction confirmation delay;

[1] The official IOTA development and operation consortium

- Based on the derived properties and metrics, we found that the real IOTA tangles present different topological features (e.g. site in-degree distributions). More importantly, we observed that the actual transaction confirmation time shows higher latency, which is not as usual beliefs that IOTA can provide much faster transaction rate than traditional blockchain.

In general, to the best of our knowledge, we are the first trying to present such an in-depth study where we publish all our source code for this empirical data analysis online[2].

3.2 Related Work

In [Kus17], this pioneering work was the first to simulate the development of the site CW values in time. However, this work is the first work to mimic an IOTA network with limited sizes as well as insufficient parameters. In [KSG18a], it further analyzed impacts of two different TSAs to site CWs and the number of tips in the tangle in a continuous-time model. However, the random walk depth is too low compared to the TSA random walk depth of 5000 used in real IOTA. The same team in [KG18] studied the relationship between the so-called *Probability of Being Left Behind*, the coefficients of walking randomness and the transaction arrival rate. A similar deficiency is also the limited size of the simulated tangles, comparing with the million-site scale in the real IOTA.

In [BRP18], an IOTA network is simulated as a multi-agent system by Netlogo, a simulation environment [WR15]. Based on the simulation, the work concludes that IOTA overcomes the shortcomings of traditional blockchains and shows both faster confirmation speed and lower computation requirement. However, the experiment is largely simplified with small sizes of the synthetic tangles and some idealistic assumptions. Besides, in [FKCM19a], an offline IOTA network was deployed to evaluate the performance by simulating some more realistic assumptions. The derived conclusion is that IOTA presents good scalability in terms of transaction confirmation rate, increasing accordingly. However, the provided results rather suggest a steady transaction rate, even if more resources are dedicated to IOTA.

In [GRW20], TSA performance in blockchains based on DAGs was evaluated. This work is also based on simplified settings such as shallow TSA random walk depth. A self-defined approval time was used to measure the confirmation rate. However, this definition is aligned neither with the real IOTA case nor with the theoretical definition from the IOTA whitepaper [Pop16]. This may yield a wrong interpretation of the actual performance in deployed IOTA networks.

In summary, the fundamental difference of this work to the state of the art is that it uses neither simulation to mimic the behaviors of IOTA for statistical analysis, nor any offline

[2] https://github.com/goldrooster/IOTA-Empirical-Data-Analysis

deployment for performance evaluations. We emphasize that the main objective of this work is to understand the nature of the real-world IOTA by statistically analyzing the original ledger data kindly made publicly available by the IOTA Foundation.

3.3 Methodology

3.3.1 Motivation

The key factors that make the IOTA mainnet different from a simulated IOTA system are:

- *Transaction Arrival Rate*: It is usually modeled as a Poisson distribution controlled by a parameter λ. However, from other online real-world blockchains do not only follow a Poisson distribution [LPDG18]. A different transaction arrival pattern influences the order of transaction attachment, which could further influence the topological features in the resulting tangle topology.
- *TSA Options*: Simulation-based studies rely on a simplified and unified random walk-based TSA assumption. In contrast, in the IOTA mainnet, TSA is not limited to one common option. Instead, various strategies are used in practical situations. For example, URTS, MCMC with customized parameters or directly referring to the Coordinator (COO)-issued milestone are acceptable choices.
- *COO Intervention*: This could be the most critical factor. The COO keeps issuing milestones, which dominates several parameters of the transaction confirmation performance. Moreover, milestones act as a special type of sites in the tangle and could affect the tangle topology properties in IOTA mainnet.

Given these differences, the question arises, whether IOTA mainnet performs anywhere near the observations in the prior art, and, in particular, to which extent COO - not considered in the prior art - affects the transaction confirmation delay. This question is the main motivation for this study, which seeks to characterize the IOTA performance using the available empirical data.

3.3.2 Tangle Reconstruction

Transaction data are being regularly collected from the IOTA mainnet by IF and published online[3]. The ledger data are archived periodically (every two or three months). The archiving activity is called *generating a Mainnet Snapshot (MS)*, wherein transactions are frozen and account balances are settled. Then, a new archive period starts. A MS mainly contains transaction records including issued milestones from COO and an approvee list that contains key-value pairs, whose key is a hash value of a transaction, and value fields contain the hash values of its direct approver transactions.

[3] http://dertangle.iota.cafe/

The first obstacle is that the published ledger data are represented in trytes but compressed in bytes. Therefore, decompression and data format conversion have to be done at the first place. After the data conversion, snapshot datasets are converted into human-readable format and saved as JSON files.

The next challenge is that tangle topology information is not explicitly kept in the datasets when generating every MS. This means that tangle topology has to be reconstructed manually. Given a MS, we have to iterate all transaction records to identify their edges and connected sites according to the hash values of the two referenced transactions (sites). A more challenging case is that multiple tangles could exist in one MS. This further requires us to manually identify the first and the last sites in order to determine one sub-tangle instance.

We provide an overview of the published MS datasets in Table. 3.1. Note that IF did not officially publish MS anymore after April 2019.

MS Index	Date (month)	Tangle#	Site# (million)	Average Site# (million)
1	2016.11	1	0.043	0.043
2	2017.01		0.115	0.115
3	02		0.09	0.09
4	06		2.5	2.5
5	08		3.5	3.5
6	09		2.1	2.1
7	10		1.2	1.2
8	2018.01	8	4.3	0.55
9	04	4	9.6	2.4
10	07	9	15.4	1.7
11	09	26	19.6	0.7
12	12	20	49.1	2.4
13	2019.04	22	43.5	2.0
Total	28	96	151.280210	1.575835

Table 3.1: IOTA Mainnet Snapshot (MS) Overview

3.3.3 Property Extraction

Given the reconstructed tangles, we then characterize their properties. First of all, we study typical graph-theoretical properties (e.g., diameter, vertex in-degree, etc.) of the tangles. Furthermore, we also calculate specific IOTA properties, e.g., site CW related to TSA. To determine the actual confirmation time, we identify the earliest milestone that approves a considered transaction from the tangle.

The challenge here is that most properties are not directly available but rather have to be calculated from the tangle. Among them, the most difficult one is to compute site

CW values. The reasons are as follows. A site CW is computed on the fly during TSA random walk procedure in the transaction attachment stage, as described above. Ergo, every attachment of a new site into the tangle may change the CW values of preceding sites. This means that the site CW value is not a static value, thus it is not provided with the published data. Calculating site CW is a graph traversal problem according to the definition of CW. Given a n-vertex graph and the graph traversal complexity $O(n)$, thereby $O(n^2)$, there is a significant computational effort given that tangle size n equals to 1.57 millions on average as per Table 3.1.

In addition, the actual transaction confirmation time is also not readily available. The main effort is on identifying the earliest milestone that approves a site. To do this, we first have to visit every milestone site in a tangle and identify all its preceding sites; after that, we calculate the time interval between the site issuing timestamp and the milestone timestamp, which tells the actual confirmation time of the corresponding transaction in the real-world IOTA. Provided that every tangle contains millions of sites, this also takes quite a long processing time.

3.3.4 IOTA Network Simulator

To facilitate our comparisons, we also use a network simulator, *TangleSimulator* [4], to generate simulated tangles, whenever necessary. There are two main parameters for tuning the simulation process. The first one is transaction arrival rate denoted as λ, and the second one is a coefficient α influencing the TSA random walk procedure. To align with the previous work [KG18, KSG18a], we choose $\lambda = 10$ but vary the value of α, and generate 10 simulated or *synthetic* tangles, each of which contains 1 million sites without particular notes.

3.4 Analysis Results

Based on the reconstructed 96 tangles, our statistical analysis results are reported here.

3.4.1 Topological Property

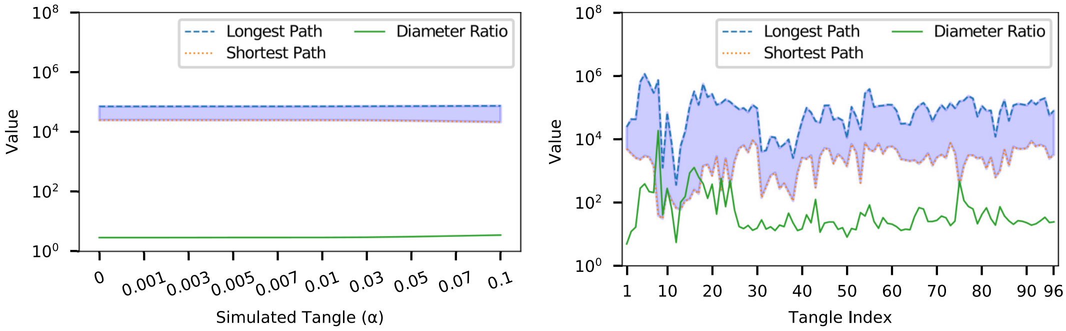
The first part of the analysis is based on graph theory, where a set of graph properties are investigated for both synthetic tangles and MS tangles.

Tangle Size

We examined characteristics of the shapes of the tangles. We first calculated the shortest and longest paths of every tangle. After that, we calculated the ratio of the two paths (called diameter ratio). The results are shown in Figure 3.1.

The diameter ratio of the simulated tangles (Figure 3.1a) is much smaller than the case of MS tangles (Figure 3.1b). In other words, the shape of simulated tangles looks closer to

[4] https://github.com/minh-nghia/TangleSimulator



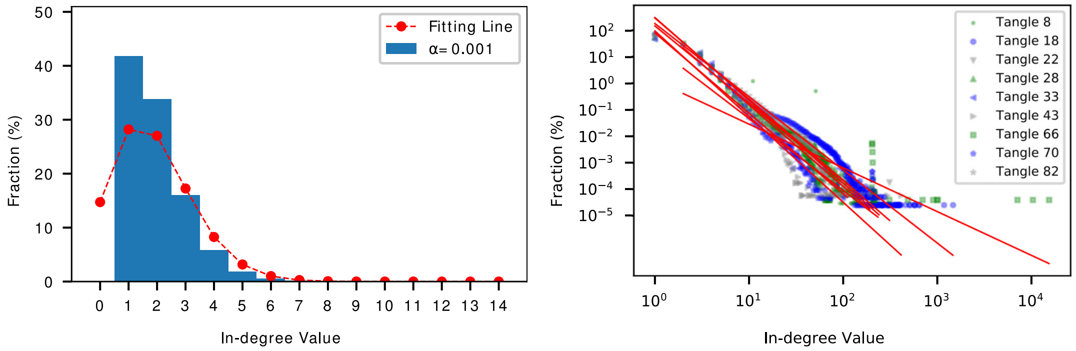
(a) Simulated Tangle

(b) MS Tangle

Figure 3.1: Ratio of Tangle Longest and Shortest Paths

a square shape with relatively equal lengths of the longest and shortest paths. However, the shape of MS tangles appear more like a narrow band shape. Another interesting point is that the shape of MS tangles seems irrelevant to the size of MS tangles, because, although the numbers of sites of the MS tangles differ a lot, the lengths of the longest and shortest paths (the blue band height) do not change drastically.

Site In-degree



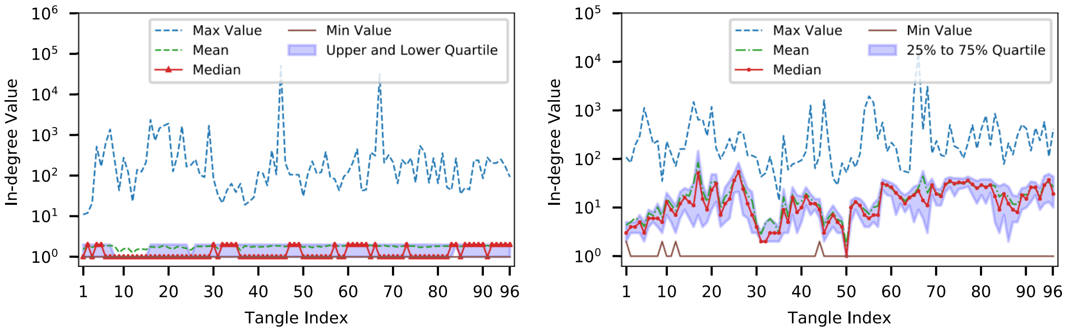
(a) Simulated Tangle (5x10^6 sites)

(b) MS Tangle (log x, y-axis)

Figure 3.2: Site In-Degree Distribution

We further generally characterize the site in-degree distribution in Figure 3.2. We first clustered the 96 MS tangles into 9 groups with k-mean clustering, where $k = 9$ and two criteria are the diameter ratio and size of MS tangles. This selects MS tangle samples that are representative enough for diversity.

Given the selected MS tangles, we plot the in-degree distribution. A key difference is that the in-degree distribution of nodes in the simulated tangles generally follows a Poisson distribution (Figure 3.2a), while the in-degree of MS tangles follows a power law distribution (Figure 3.2b), with fitted curves in shown in red respectively.



(a) Usual Site

(b) Milestone

Figure 3.3: MS Tangle Usual Site and Milestone In-degree Comparison

We are further interested in the degree features of different types of sites (e.g., usual sites and milestones). In MS tangles, the in-degree values of usual sites are small (Figure 3.3a), where mean and median values are around 1 or 2 overlapping with middle two quartiles, although there are some exceptional cases with higher degrees in the range of $[10, 10^3]$. However, for the case of milestones (Figure 3.3b), we notice that the mean and median values of milestones range between $[5, 10^2]$, overlapping with middle two quartiles. This is several magnitudes higher than the cases of usual sites. It seems that milestone sites are selected more frequently in real IOTA.

3.4.2 Specific IOTA Property

We then present the analysis on IOTA specific properties.

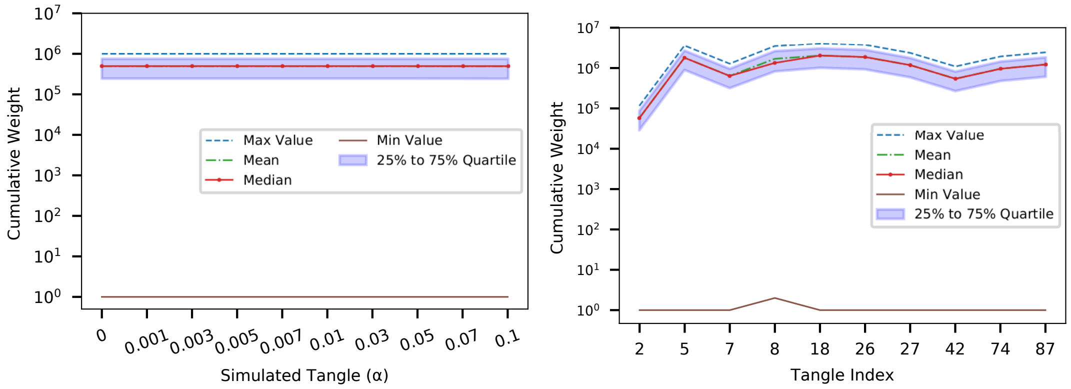
Site CW

As we explained in Section 3.3, site CW is a dynamic value calculated on the fly. Thus, these values are not present in the published MS. We have to repeat the TSA random walks to recalculate them. Similarly, we used k-mean clustering, where $k = 10$ to select 10 MS tangles in different shapes and sizes.

Site CW values of the selected MS tangles are slightly higher than the case of simulated tangles. It seems that tangle topology does not influence the CW that much. The possible reason could be that site CW is a value added up with all sites in a sub-tangle, which dissolves and normalizes the impact of topological differences.

Edge Weight

Based on site CW, we define a new edge property called EW as the absolute difference of its two sites' respective CW values. This value implies the location of a site to attach to. For example, the EW of a newly added edge shall be small, if a new site attaches to a recently attached tip, whose CW is similar to the new site. Oppositely, if a new site attaches to an old site, the EW of the newly added edge is large, because the difference

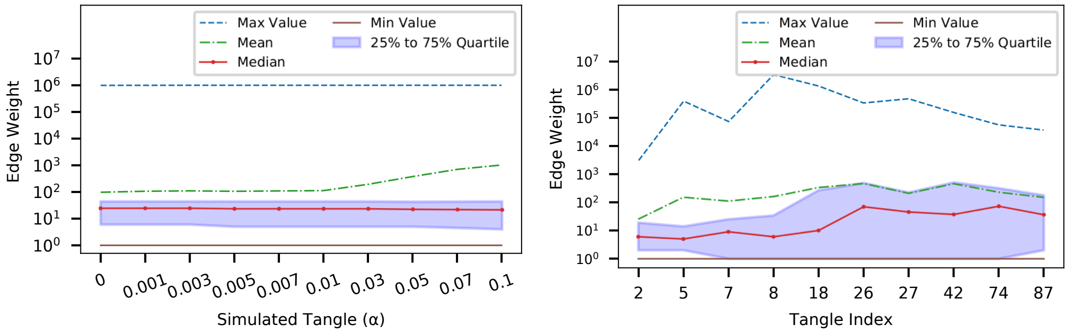


(a) Simulated Tangle

(b) Selected MS Tangle

Figure 3.4: Site CW Analysis

of the two sites' CW values is large. Therefore, EW can be an indicator of 1) a *lazy site*, which does not select a recent tip but an old site or 2) a *parasite chain* phenomenon, where a fork diverts from the main tangle. The two cases are generally called *abnormality*.



(a) Simulated Tangle

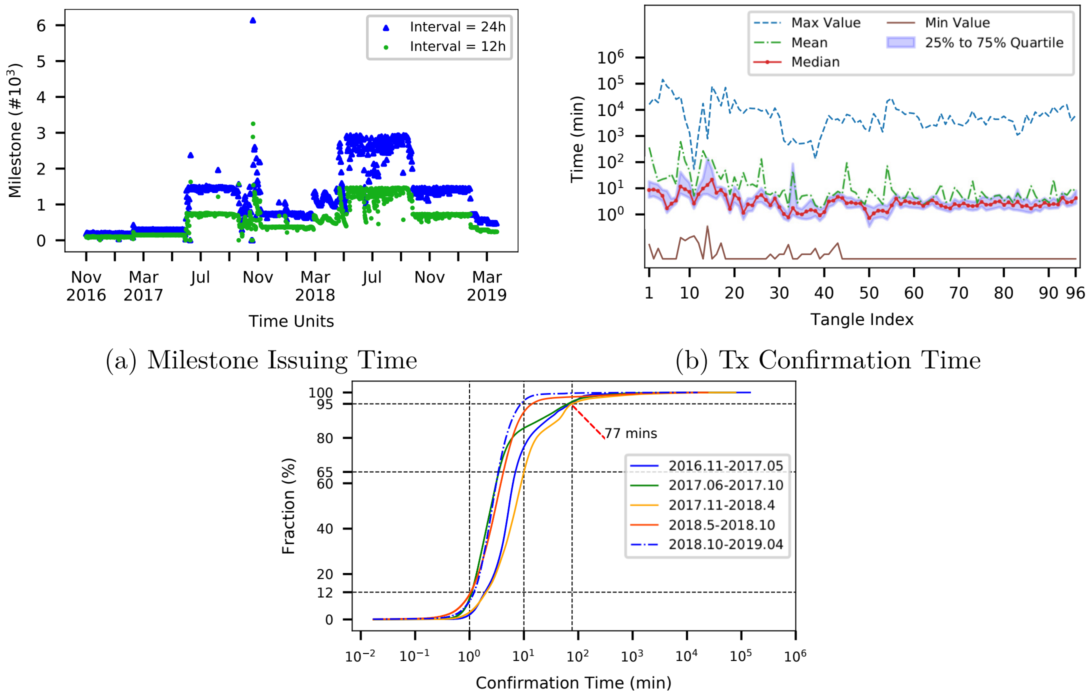
(b) Selected MS Tangle

Figure 3.5: Tangle Abnormality

Both simulated and MS tangles generally show certain amounts of abnormalities according to the results in Figure 3.5. It seems that the abnormality of simulated tangles is more stable than the case of MS tangles in terms of the variations of EW values. This might be because the simple TSA strategy is used in simulated cases, while more diversified TSA strategies are adopted in real IOTA. In general, both CW and EW are less influenced by the tangle topology.

3.4.3 Transaction Confirmation Performance

Finally, we evaluate the transaction confirmation performance based on the criteria used in IOTA mainnet. Ideally, a transaction is considered as approved, once that transaction is attached by a new coming site in a tangle. However, according to the definition of IF, in reality a transaction is considered as confirmed, only if it is approved by a milestone. In a MS tangle, this means that a milestone site directly or indirectly connects to the considered site. This might delay the confirmation time, because milestones are not always issued timely.



(a) Milestone Issuing Time

(b) Tx Confirmation Time

(c) Tx Confirmation Time CDF in 5 Periods

Figure 3.6: MS Tangle Tx Confirmation Performance

We first summarized in Figure 3.6a the issuing rate of milestones in two different scales of time intervals (every 12 and 24 hours). Before May 2018, the issuing rate fluctuated between several hundred and 1800 per day; after that, the issuing rate increased significantly up to 3000. The issuing rate slowed down since September 2018 to 500 per 12 hours and 1500 per day. It further went down to 500 per day since January 2019.

We then studied the distribution of confirmation time of transactions in all MS tangles (Figure 3.6b). The median value of confirmation time ranges around 10 minutes. This also applies to 25% to 75% quantile transactions (blue band areas). With some exceptional cases, the confirmation time gets maximum value ranging between $[10^2, 10^4]$ minutes. This also stretches the mean value above the median value curve. Another observation is that before November 2018 (MS Tangle 55), the confirmation performance in real IOTA had larger fluctuation and became more stable after that. It is worth noting that the maximum time range ($[10^2, 10^5]$) is a confirmation time delayed from 1.6 hours to 6.9 days. In the light of the IoT orientation, this would rather seem as a considerably long transaction delay.

We further investigate cumulative proportions of transactions that are confirmed after a certain duration. We divided the whole period, since IOTA mainnet was launched into five periods with a half-year step. Statistically, we calculated the Cumulative Distribution Function (CDF) of transaction confirmation times of the five periods in Figure 3.6c. In the presented results, we found that the transactions that are confirmed in less than 1 minute are rather very few, at max. 12%, with two of the periods exhibiting values lower than roughly 5%. The confirmation rate increased rapidly in between 1 and 10 minutes, where the proportions of confirmed transactions reached at least 65% in two periods and the other three periods even reached more than 85%. It took 77 minutes for all periods to confirm 95% of transactions. However, almost every period has a small proportion (around 1% to 5%) of transactions that were delayed for an exceptionally long time.

3.4.4 Key Observations

1. Real IOTA generates tangles with different topological features, compared to the simulated tangles. The shape of the real tangles is narrower. Real IOTA tangles show a power-law degree distribution rather than a Poisson distribution as in simulated cases.

2. Nodes in the real IOTA indeed use various TSAs to attach new sites into their local tangles. Milestones are selected more often than usual sites. Abnormal sites were observed in reality (cf. the result of EW Analysis), which are not simulated in most of the prior art, where nodes perfectly follow the IOTA specification;

3. The transaction confirmation rate is not as high as usual believed, if the confirmation by a milestone is required. This needs the assistants from milestone sites. Because of that, most of transactions (> 50%) are confirmed in around 10 minutes, and there is a small proportion of transactions delayed for several days. The normal confirmation time in IOTA mainnet seems equivalent to the performance in typical traditional blockchains (i.e., having to wait roughly 10 minutes, until the transaction can be considered confirmed). This is far behind the requirement to support lightweight, rapid and instant IoT applications that are delay-sensitive, especially considering those exceptionally delayed transaction cases.

3.5 Summary

In this chapter, we provide an in-depth analysis on the real IOTA tangle based on historical empirical data from the IOTA mainnet. We reconstructed the tangles from the empirical ledger data, analyzed the tangle properties and presented a comprehensive statistical analysis. According to the presented results, our key findings are that the features of the real IOTA tangles are topologically different from the simulated tangles; more importantly, the transaction confirmation time largely depends on the milestones issued by COO. In addition, it is inefficient to rely on the mechanism of using site cumulative weight in the random walk of TSA, which aligns with the recent plans of the IOTA Foundation. We hope that the presented results can provide a better understanding of the nature of the real IOTA and motivate to continue further analysis in the IOTA community.

CHAPTER 4

Modeling Ledger Dynamics in IOTA Blockchain

IOTA blockchain is a new type of distributed ledger systems that is lightweight without mining and feeless-of-using. Rather than using a chain structure as in traditional blockchains, IOTA organizes ledger records with a DAG, called Tangle. When message entries are committed into the ledger, the ledger tangle grows in a special way where multiple messages could be attached by different processing nodes in parallel. Such a unique evolution process motivates us to study the ledger tangle dynamics, which is unexplored so far. In this chapter, we present the first generative modeling for IOTA tangle based on stochastic analysis. A key finding is that IOTA tangle renders a dPLN distribution, rather not typical network models (e.g., Power-Law and Exponential distributions). Quantitative comparisons show that the fitting quality of our model outperforms existing popular models on official real-world datasets published by IOTA Foundation. Estimated model parameters are provided, which is immediately instrumental for a more realistic IOTA network generator design. The proposed generative model also provides a deeper understanding of the internal mechanics of IOTA network.

The structure of this chapter is outlined. We review existing network models in Section 4.2 and introduce IOTA preliminary as a background in Section III; Section 4.3 presents our model and Section 4.4 introduces model fitting; after that Section 4.5 shows the comparison results with existing popular models; Section 4.6 concludes this chapter.

4.1 Introduction

In 2016, IF proposed a new type of blockchain–IOTA network. Instead of using a chain topology, IOTA uses a DAG, called Tangle topology, to organize ledger data on every processing node [Pop16]. IOTA abandons PoW consensus and is feeless. Thus, IOTA is suitable for many IoT applications, where communications can be characterized as instant, massive exchange of tiny messages. Although IOTA enjoys a high popularity in research, most of the studies focus on statistical analysis [FKCM19b], protocol extension/enhancement [PAD20] and applications [BVF18b], [XGH20]. In contrast, we would like to gain more insights into theoretical analysis, notably through network graph modeling, to reveal and better understand the core mechanism of IOTA network.

IOTA tangle evolves in a special way. Specifically, every vertex represents a single message record in the ledger (either a value transaction or a simple data payload). For a new message (e.g., submitted by a user), it will be attached as a new vertex with introducing new directed edges to existing vertices in the tangle. Semantically, each directed edge represents the approval from the source vertex (message) of the target vertex (message). Vertice selection is determined by specific selection algorithms, defined by the distributed consensus protocol. The key idea is to stimulate processing nodes attaching new messages biasing on tips, i.e., vertices having no approver yet thus their in-degrees are zero. With the new vertices independently added in the tangle by different nodes, the size of the tangle will grow with multiple vertex- and edge-arrivals over time.

In this chapter, we are interested in the tangle ledger dynamics driven by IOTA.

This chapter focuses on the tangle ledger dynamics driven by IOTA. It would be ideal to develop a formal network model capable of correctly describing the evolution of an operational IOTA tangle and in particular its stationary degree distribution. Alas, it appears unlikely that usual network models, e.g., random graphs [ZMN17, ACL01, Gar09]or the Barabasi-Albert's Preferential Attachment (PA) model [BA99] can correctly explain the IOTA tangle behavior. Key differences are as follows.

First of all, IOTA tangle typically grows non-uniformly, in bursts, during which multiple new vertices are added at the same time, each with more than one new edge. The reason for such bursts is tangle consolidation: since every node independently attaches incoming messages to its local tangle ledger copy, at one point, those individual tangles need to be consolidated and merged to one. During this phase, multiple messages and edges are added in one batch. In contrast, prior work usually assumes a single node arrival mode, and very often simplifies the process further to single edge addition. Hence, it is inaccurate to simplify IOTA tangle growing with a single vertex arrival mode.

Secondly, vertex and edge additions do not follow a simple PA model (or any of its variants). In PA model, a vertex is randomly selected proportionally to its degree. In IOTA, however, tip selection is a distributed decision-making process to identify a valid branch, where a new vertex can safely attach without causing conflicts. Such a process involves evaluating other existing vertices (i.e., historical messages) in a sub-tangle topology, thus it cannot be trivially abstracted as a simple vertex attribute (e.g., a degree value) as in PA model.

We will see that an alternative is required in order to derive a network model that can capture the essence of the ledger dynamics. In summary, the main contributions of this chapter are as follows:

- We employ stochastic analysis to characterize IOTA tangle evolution with an SDE that can approximately govern the vertex degree dynamics over time;
- We discover that operational IOTA tangles can be accurately described through the dPLN distribution;
- We quantitatively compare fitting quality of the proposed model against other popular candidate models with the real-world data (whose size is around 320G) published by IF. The results confirm the correctness of our key finding, where IOTA tangles render a dPLN degree distribution.

To the best of our knowledge, this is the first theoretical work modeling IOTA network dynamics. Estimated model parameters in this work are immediately instrumental for a more realistic IOTA network generator design.

4.2 Related Work

Though there are very limited relevant theoretical works, we observed several attempts on analytical performance modeling about IOTA. In [Kus17], they built a rule-based discrete model and a continuous-time model for IOTA, respectively, in order to build the relationship of the number of tip vertices and the vertices' cumulative weights over time. In [PSF19b], the authors analyzed the message attachment behavior of IOTA network and proved that there exists a Nash equilibrium, revealing that selfish nodes will cost more than non-selfish nodes. This work targets to a different goal, which aims to theoretically model how the tangle topology evolves and what a degree distribution could best represent it.

The main difference of the IOTA network to random graph models summarized in [ZMN17] is that IOTA's ledger tangle is growing, while random graph models consider graph's size unchanged. This motivates us to consider those models characterizing evolving graph networks.

A famous growing/evolving network model (i.e., PA model) was proposed in [BA99]. In this model, new vertices attach to target vertices selected proportionally to their degree (often periphrased as "rich gets richer"). The authors have shown that applying this simple principle results in a scale-free network, i.e., a graph with a Power-Law (PL) degree distribution). Hence, PA is a popular generator for (a particular class of) scale-free networks.

Cyclic PA (CPA) was introduced in [KP13] as a variant of the PA model. In CPA model, the attachment probability depends on the shortest path from the node to all other nodes. The author used this model to analyze the real world network, such as online social networks and relations amongst company leaders. The finding is that the proposed CPA provides more flexibility to model the real life networks. Additionally, [WGYZ09] proposed another PA model variant to model a phenomenon, where a vertex acquires a new vertex depending on the density of its local area in a graph. Authors analytically obtain stable degree distributions and cluster in-degree correlations. They show the emergence of a PL distribution of the resulting graph's degrees.

Although PA-like models provide decent modeling for a large number of evolving networks, the message attachment in IOTA behaves differently. One key difference is that most of PA models only consider a single vertex arrival mode, while IOTA tangle grows with a batch arrival mode. Another key difference is that the attachment probability is determined by running an algorithm applied on a sub-tangle topology, which cannot be written in an analytical form as in PA models.

In reality, many phenomena are not following PA models. Their degree distributions are also not PL/Exponential (Exp) distributions . For example, the authors showed respectively that the file size [Mit03a], the city size [siz] and mobile call graphs [SMS+08a] follow dPLN distributions [RJ04a]. Compared to them, the IOTA network is a distributed system and ledger dynamics are implicit. Hence, a correct modeling is required.

4.3 IOTA Tangle Ledger Dynamics

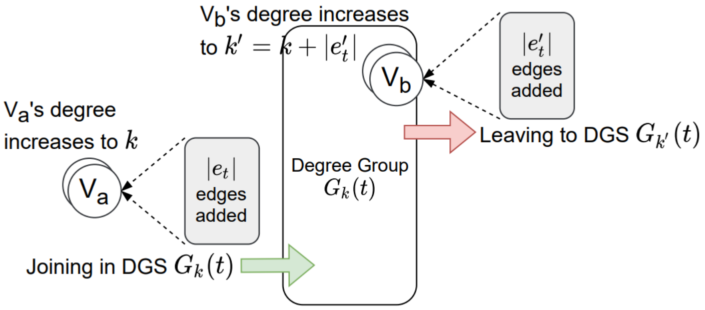


Figure 4.1: Dynamics of Degree Group Size (DGS) $s_k(t)$ of $G_k(t)$

4.3.1 Key Idea

We denoate a subset of vertices in a tangle where all vertices have the same degree $k \in \mathbb{Z}^+ \cup \{0\}$ at time t, called a *degree group*. The cardinality (size) of a degree group is $|G_k(t)| = s_k(t)$, called DGS. Let us further denote all new edges, which are added from several new messages to the same vertex, as e_t, called an *edge group*. In IOTA network, $s_k(t)$ cna change because of the following two ways as illustrate in Figure 4.1.

The first way is that $s_k(t)$ increases, because there could be one or more vertices, whose original degrees are less than k but an edge group e_t makes the degree increase to k by adding $|e_t|$ new edges. V_a in Figure 4.1 is such an example, where V_a joins in degree group $G_k(t)$ and increase $s_k(t)$ by one.

The second way is that $s_k(t)$ decreases, because there could be a vertex, whose degree is already k, but another edge group e'_t makes its degree increase to k' by adding $|e'_t|$ new edges. V_b in Figure 4.1 is an example, where V_b leaves to degree group $G_{k'}(t)$ and decreases $s_k(t)$ by one.

Mapping to IOTA network, the first situation happens to any vertex with a degree value between $[0, k)$, and the second situation happens to any vertex with a degree value is equal to k, thus covering all types of vertices in a tangle. For instance, a tip vertex (i.e., degree value is 0) will join in degree group $G_k(t)$, if an edge group et adds exactly k edges to it (i.e., $|e_t|$ is k); any non-tip vertex whose degree is $< k$ will also join in degree group $G_k(t)$ if an edge group et adds up its degree value to k. However, attaching to any vertex, whose degree is already k will make the vertex leave the degree group $G_k(t)$.

From a statistical view, the macro effect of the joining and leaving vertices of a degree group $G_k(t)$ can be viewed as a Brownian motion [Nel67], because how DGS $s_k(t)$ will exactly change is a stochastic process, which is driven by the random behaviors (e.g., random vertex selections) from processing nodes in IOTA blockchain.

4.3.2 A Stochastic Model

Considering the above two possible ways $s_k(t)$ may change, the ratio of the variation of $s_k(t)$ to its original value $s_k(t)$ can be either positive, zero or negative. Mathematically, it can be formulated with a SDE of DGS $s_k(t)$ as follows.

$$\frac{ds_k(t)}{s_k(t)} = \omega(t)dt + \sigma(t)dB(t) \tag{4.1}$$

where $\omega(t)$ and $\sigma(t)$ are coefficients characterizing the growth rate of DGS and the variantion of DGS resulting from random selection behaviors, which is modeled as a Brownian motion $dB(t)$. Another implicit necessity is that $s_k(t)$ must be non-negative value. However, if we directly model the amount change of $s_k(t)$ (rather than the ratio $\frac{ds_k(t)}{s_k(t)}$ as in Equ. 4.1), the Brownian motion term may lead to $s_k(t)$ becoming negative, which would contradict its definition.

The SDE in Eq. 4.1 agrees the form of Geometric Brownian Motion (GBM), which is analytically solvable if $\omega(t)$ and $\sigma(t)$ are time independent. Interested readers are referred to [KS98] for the details of deriving GBM's theoretical properties. Here we recap them as follows:

1) The solution of the SDE of sk(t) in Eq. 4.1:

$$s_k(t) = s_k(0)exp(\underbrace{(\omega - \frac{\sigma^2}{2})t}_{Denoted\ by\ \mu} + \sigma B_t) \tag{4.2}$$

   where the μ term is used in the following equation.

2) The Probability Density Function (PDF) of $s_k(t)$ at any observation time t follows a Lognormal (LN) distribution:

$$p_{LN}(x,t) = \frac{1}{x\sigma\sqrt{2\pi t}} exp(\frac{-(logx - t\mu)^2}{2t\sigma^2}). \tag{4.3}$$

3) The PDF of $s_k(t)$ at an exponentially distributed observation time t (i.e., $p_T(t) = \xi e^{-\lambda t}$) follows a dPLN distribution:

$$\begin{aligned} p_{dPLN}(x) = \frac{\alpha\beta}{\alpha+\beta}[x^{-\alpha-1}A(\alpha)\Phi(\frac{logx - \mu - \alpha\sigma^2}{\sigma}) \\ + x^{\beta-1}A(-\beta)\Phi^c\frac{logx - \mu + \beta\sigma^2}{\sigma}], \end{aligned} \tag{4.4}$$

where $A(z) = exp(z\mu + \alpha^2\sigma^2/2)$, $z = \alpha, -\beta$, $\Phi(\cdot)$ and $\Phi^c(\cdot)$ are the CDF and complementary CDF of a standard normal distribution, respectively. The model parameters of a dPLN distribution are $\Theta := [\mu, \sigma^2, \alpha, \beta]$, which will be estimated from observed data.

The interpretation to our problem is that the DGS $s_k(t)$ grows along with the tangle over time t and the stoppage time t is assumed exponentially distributed. Importantly, the PDF in Equation 4.4 tells what the probability density the size of a certain degree group $G_k(t)$ will be. After normalized with the total tangle size n, it tells exactly the degree distribution of a tangle that we target.

4.4 Model Fitting

4.4.1 Fitting Data Preparation

We use real ledger data generated from IOTA mainnet on Internet that are published by IF. The whole dataset contains ledger records from 2016.11-2019.06 (Period I) and 2020.04-2020.08 (Period II). Period I contains 96 tangles and Period II contains 16 tangles (112 tangles in total). The number of messages of reconstructed tangles vary from several thousands to about 40 millions. To prepare the data for model fitting, there are two main challenges when processing the original datasets as follows.

The first challenge is that the published ledger data is represented in trytes but compressed in bytes. Thus, decompression and data format conversion have to be done at the first place. After the conversion, the datasets are converted into human-readable format and saved as JSON files.

The second challenge is that tangle topology information is not explicitly recorded. This means that tangle topology has to be reconstructed manually. We iterate all message records to identify their edges and connected vertices according to the hash values of the two referenced messages. A more challenging case is that multiple tangles could exist in one batch. This further requires us to cluster messages manually that belong to the same tangle by identifying individual genesis vertices and the last vertices attached to the tangle starting from a particular genesis.

After all tangles are reconstructed, the vertex's (in-)degree value is calculated by summing up the total number of attached messages of the considered vertex. For a certain degree group G_k, its DGS s_k is the number of vertices having the same degree k in the tangle. For each vertex in $G_{k,\forall k\in[1,K]}$, the observed probability (proportion) of such a degree group $y_i = s_k/n$. This thus gives the fitting data for the proposed model.

4.4.2 Model Parameter Estimation

Maximization Likelihood Estimation (MLE) is a general method of estimating the parameters of an assumed probability distribution model, given observed data. Mathematically, this is achieved by maximizing the likelihood of observed data $\mathcal{Y}$ with an presumed parametric model characterized by parameter θ. Specifically, we have:

$$\theta^* \leftarrow arg \max_{\theta} \ell_{PDF}(\theta; \mathcal{Y}), \tag{4.5}$$

where the $\ell_{PDF}(\cdot)$ is the log-likelihood function defined ass follows:

$$\ell_{nLP}(\theta; \mathcal{Y}) = \sum_{i=1}^{n} log f_{PDF}(\theta; y_i) \tag{4.6}$$

where $f_{PDF}(\cdot)$ is the PDF of the presumed model. For example, it can be dPLN model's PDF Equation 4.4, or any other candidate models.

To solve Equ. 4.5, in the simplest cases, where an analytical solution of the optimal estimate exists, the optimal estimate can be obtained directly. This situation exists to most of simple statistical models such as PL and Exp distributions and so on. In difficult cases, where the analytical solution does not exist, solving MLE needs numerical algorithms. Since the fitting algorithm is not the main focus of this work, we follow the MLE formulation and use an optimizer 'L-BFGS-B', which is a classical gradient-descent method proposed in [ZBLN97a], to solve the MLE problem in Equ. 4.5. Note that this routine is commercialized and directly available in Python library.

Note that though MLE is a principal way to handle the parameter estimation problem, the key issue of using MLE is that the likelihood function is usually not convex or concave (due to the sum of a number of log-PDF terms). Hence, whether or not the estimated parameter is a global optimum is uncertain. In fact, there are rich research topics on non-linear optimization, which is also planned as our future work for studying the efficiency of dPLN's parameter estimation.

4.5 Results

4.5.1 Candidate Models and Scoring Metrics

Model	PDF	Model Parameters	Closed-Form
PL	ζx^{γ}	γ	Closed-Form
Exp	$\xi e^{-\lambda x}$	λ	Closed-Form
LN	Equ. 4.3	μ, σ^2	Closed-Form
dPLN	Equ. 4.4	$\mu, \sigma^2, \alpha, \beta$	L-BFGS-B

Table 4.1: Summary of candidate models (ζ and ξ: Normalization Constants)

The candidate models for comparisons and their parameter estimations are summarized in Table 4.1. The model complexity increases from PL and Exp to LN and dPLN. The number of model parameters also increases from 1 to 4, thus becoming reasonably representative for both model performance and complexity.

We choose Root Mean Sqaured Logarithmic Error (rMSLE) to measure the fitting quality of different models. Its definitionis given below:

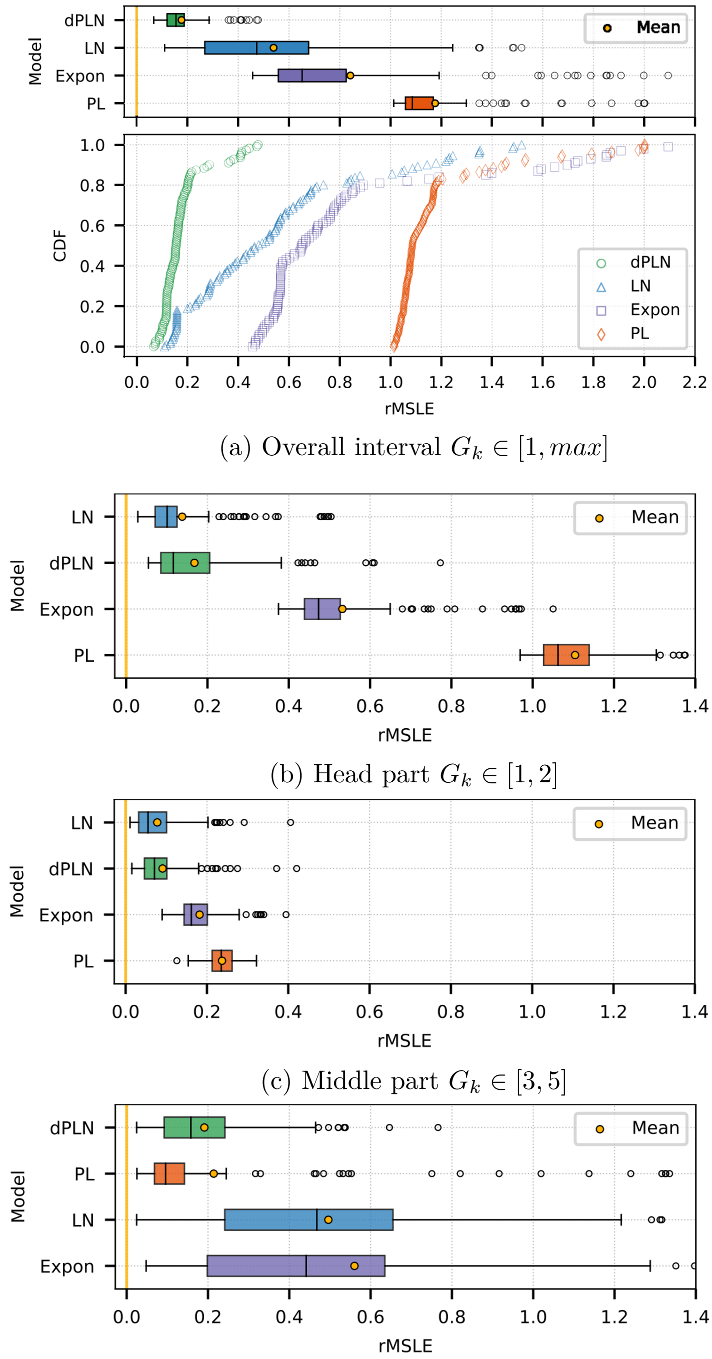
$$rMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log\, y_i - log\, \hat{y_i})^2} \tag{4.7}$$

where n is the total number of observed vertices, and $\hat{y_i}$ is the predicted probability value of observed probability value y_i. rMSLE measures the relative errors of the predicted and actual values. The reason to choose rMSLE is that the probabilities between different types of vertices may be significantly different with several magnitudes. In this case, unit dependent measures e.g. Mean Square Error (MSE) turns out to be unsuitable because the absolute distances of errors from data points with smaller proportions will be overwhelmed. rMSLE solves this issue so that it becomes unit independent by taking a log-difference/relative ratio.

4.5.2 Quantitative Fitting Comparison

Figure 4.2 shows the four candidate models' performance scored by rMSLE on different parts of the vertex population. The optimal rMSLE is 0 highlighted with a yellow bar, meaning all observed and predicted data exactly match.

On the overall interval (Figure 4.2a), the rMSLE mean of dPLN model is 0.3. LN model is at the second place but its rMSLE mean is about 0.5, which is worse than dPLN model's. Both Exp and PL are incorrect models to explain the observed in-degree distributions of the tangle snapshots with much larger rMSLE.



(a) Overall interval $G_k \in [1, max]$

(b) Head part $G_k \in [1, 2]$

(c) Middle part $G_k \in [3, 5]$

(d) Rear part $G_k \in [6, max]$

Figure 4.2: Comparisons of candidate models with rMSLE

Parameters	μ	σ^2	α	β
Mean	0.29	0.15	17.00	14.92
Variance	0.03	8e-3	517.93	185.28
1/4 Quartile	0.16	0.09	2.32	5.01
Median	0.32	0.18	3.17	9.65
1/4 Quartile	0.44	0.21	36.21	27.72

Table 4.2: Statistics of estimated model parameters

On segmented parts, in the head and middle parts, LN model performed slightly better than dPLN model (see Figure 4.2b and Figure 4.2c). However, the rMSLE mean shown by two models are actually quite close to each other, especially the median rMSLE. Neither Exp nor PL models fits these two parts well, especially in the head part. In the rear part (see Figure 4.2d), the best is dPLN model. Surprisingly, the performance of LN model degrades significantly, although it performs well on the previous two intervals even slightly wins against dPLN model.

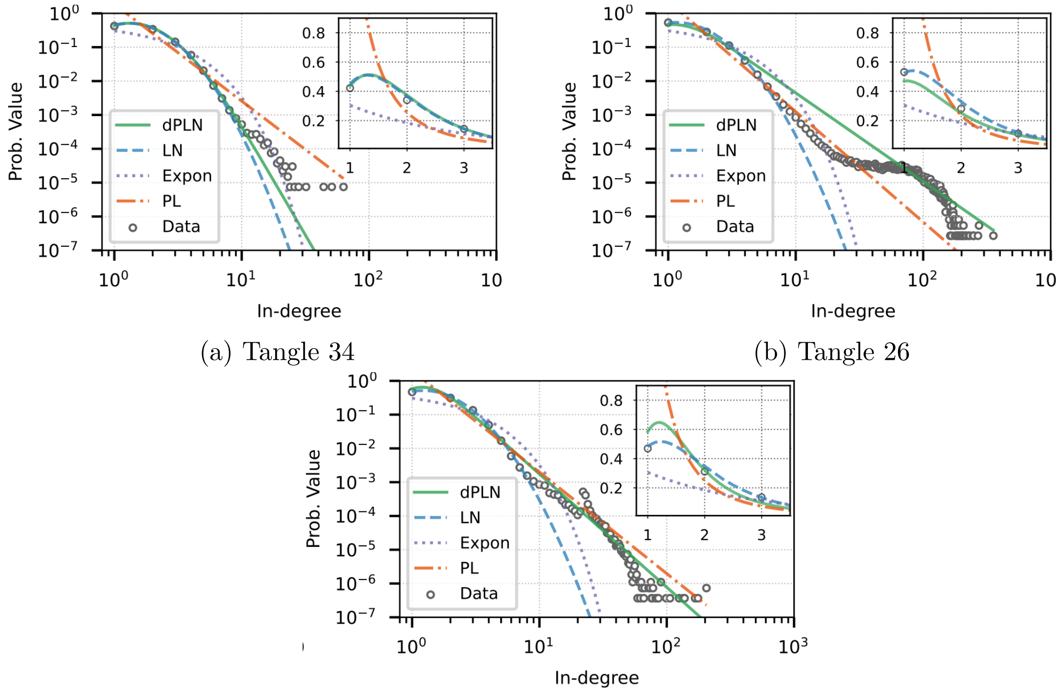
As we know, the uniqueness of a population is determined by the minority instead of majority features. The segmented comparisons above justify this fact because although a candidate model can perform better to some common features, its overall performance can still be hindered. For example, LN model strongly biases to fit vertices in the head and middle parts, in which both are majority. However, LN model completely ignores the minority feature of higher degree vertices in the tangles. Although the proportion of high degree vertices is small, a significant divergence on them failed LN model's overall performance. In contrast, only dPLN model showed a good balance between majority and minority features, which explains why it can eventually achieve an overall quality fitting results.

We also provide the estimated values of model parameters in Tab. 4.2. These values can be directly used with our model to generate tangles that give the most realistic topology as in IOTA mainnet.

4.5.3 Graphical Fitting Comparison

We then provide a graphical comparison of the candidate models with three tangle examples in Figure 4.3. This helps readers to capture the difference of model performances in a visual way. For fairness, we pick the three tangle samples with top 25%, median and bottom 25% rMSLE of dPLN model, respectively. We also zoom in the fitting of degree group [1, 3] in the subplots at the upper right corner.

Generally, the graphical fittings match the quantitative results. Specifically, dPLN model (green-solid curves) fits averagely closer to the observed distributions in all parts. Additionally, LN model fits slightly better to the head part but extremely poorer in the rear part. As we can see, it diverts the farthest to the tails. Moreover, none of PL and Exp models is a reasonable choice.



(a) Tangle 34

(b) Tangle 26

(c) Tangle 61

Figure 4.3: Graphical fitting comparison

In summary, the evaluation results can tell that with the real-world data from IOTA mainnet, the network dynamics result in a dPLN distribution, which invalidates the typical assumption of either PL or Exp models in this space.

4.6 Conclusion

In this chapter, we modeled IOTA ledger dynamics with stochastic analysis and analytically derived its degree distribution. Our key finding is that the tangle topology of IOTA network renders a dPLN distribution. This finding was confirmed by fitting our model predictions to official datasets and the proposed model outperforms other existing models. We hope that this promotes a deeper understanding of the mechanism of IOTA and hence benefit IOTA network generator design for further research and/or application purposes.

CHAPTER 5

A Theoretical Model Characterizing Tangle Evolution in IOTA Blockchain Network

IOTA blockchain system is lightweight without heavy proof-of-work mining phases, which is considered a promising service platform of Internet of Things applications. IOTA organizes ledger data in a DAG, called Tangle, rather a chain structure as in traditional blockchains. With arriving messages, IOTA tangle grows in a special way, as multiple messages can be attached to the tangle at different locations in parallel. Hence, the network dynamics of an operational IOTA system would justify a thorough study, which is currently unexplored in the literature. In this article, we present the first theoretical modeling for the evolving IOTA tangle based on stochastic analysis. After analyzing snapshots of the real-world IOTA ledger data, our key finding suggests that IOTA tangle follows a rather atypical dPLN degree distribution. In contrast, typical power-law and exponential distributions do not accurately reflect the fact. For model parameter estimation, we further realize that using generic optimization solvers cannot yield quality fitting results. Thus, we design an alternative algorithm based on the EM framework. We evaluate the proposed model and fitting algorithm with official data provided by the IOTA Foundation. Quantitative comparisons confirm the fitting quality of our proposed model and algorithm. The whole analysis reveals a deeper understanding of the internal mechanism of the IOTA network.

The remainder of this chapter is organized as follows. Related work is reviewed in Section 5.2; Section 5.3 presents our model and Section 5.4 introduces our model fitting algorithm; after that, Section 5.5 shows the evaluation results; Section 5.6 concludes this article.

5.1 Introduction

In 2016, IOTA Foundation - The official IOTA development and operation consortium—launched a new type of blockchain system, called IOTA. Rather than using a chain topology, the ledger of the IOTA system is organized as a DAG, called Tangle, wherein every vertex represents a single message record (either a value transaction or a data payload) [Pop16]. In IOTA, every participating node holds a copy of the tangle, responsible for committing incoming messages independently and forming consensus in a distributed manner among participants.

IOTA is lightweight and feeless without heavy proof-of-work mining phases. Hence, IOTA is considered suitable for a decentralized service platform of IoT applications, characteristic to a massive exchange of instant, typically tiny information. Although IOTA recently has gained high research popularity, most of the works focus on its empirical studies [FKCM19b], protocol extensions [PAD20], and applications [BVF18b], [XGH20]. Nevertheless, we are not aware of any graph- and network-theoretical analysis on the ledger tangle evolution, especially for the operational IOTA network deployed in the real world. Undoubtedly, theoretically understanding how the tangle evolves is important to capture the core mechanism underlying IOTA network dynamics.

Due to the particular structure and the distributed consensus mechanism, the evolving ledger tangle in IOTA would justify such a network dynamics analysis. Specifically, when a new message arrives, it is attached as a new vertex, with directed edges pointing to the existing vertices. As in a DAG, many candidate vertices exist, the location the vertex will attach to is determined by a selection algorithm—part of the IOTA distributed consensus protocol. In IOTA, a directed edge represents an approval from the source vertex to the referred vertex. The key principle is to encourage new messages approving yet unapproved vertices, so-called tip vertices, whose in-degree is zero. The ledger tangle chronologically grows in such a manner over time. More details about IOTA's mechanism will be introduced in a later section.

This chapter focuses on how the tangle topology evolves in IOTA. Particularly, we try to answer, if there exists a theoretical network model governing this process; and if so, what a degree distribution would best represent it. Several typical network models, such as the random graph model [ZMN17], [ACL01], [Gar09] and Barabási's PA model [BA99] have gained a wide recognition, after they were shown to have good fitting properties for many naturally occurring processes. However, during our initial investigation, we realized that the existing network models did not fit well with the observed data sets generated from the IOTA network. The key reasons are explained as follows.

First, IOTA tangle grows with a batch arrival mode, in which multiple new vertices may come and every new vertex may add more than one new edge. The key fact behind this is that a copy of the tangle exists on every participating node; and every node can attach new messages to its local tangle independently; thus, after individual ledger copies are merged, multiple messages and edges can appear to one vertex at burst. Existing models, however, often assumed a single vertex sequential arrival mode, where only one new vertex is added at each time. Most even further assumed a single edge addition. Hence, it is inaccurate to simplify IOTA tangle developing with such a simple way. The network modeling for IOTA's tangle evolution has a different growing behavior.

Second, vertex and edge addition in IOTA tangle do not follow a similar logic of the PA model (or of its variants). In the PA model, a vertex is randomly selected proportional to (or modeled as a function with explicit form of) its degree value. In IOTA, however, vertex selection is a much more complicated process, which involves evaluating other existing vertices (i.e., historical ledger records) in a subtangle topology. Clearly, it cannot be attributed to a simple vertex property (e.g., a degree value) characteristic to the PA model. The above two key features of the formation process makes IOTA's tangle evolution show a unique behavior, which was not studied in the scope of network modeling research.

In addition to deriving the model, another technical problem that is equally important is parameter estimation. Unfortunately, the issue we encounter is that a generic optimization solver (typically Gradient Descent (GD)-based methods) cannot give a satisfactory parameter estimation for the derived model. We then develop a dedicated algorithm based on EM framework [DLR77] as an alternative. In summary, our main contributions are listed as follows.

1) Using stochastic analysis, we characterize operational IOTA network dynamics with an SDE that approximates vertex degree evolution over time.
2) Based on the analytical solution of the modeled SDE, we derive that IOTA tangle dynamics follow a dPLN distribution.
3) For parameter estimation, we develop an EM-based algorithm, which can provide more reliable and higher. quality fitting results than using generic GD-based solvers; our source code is also published to benefit the community.
4) We evaluate the fitting quality of the derived model and proposed algorithm with realistic snapshot data generated from IOTA mainnet, and the results justify our findings.

To the best of our knowledge, in short, this work is the first trying to model the tangle evolution in IOTA, whose network dynamics behaviors combine a batch vertex arrival and a complex attachment process. However, modeling of such a unique network dynamic, meanwhile providing a more efficient fitting algorithm, were not seen so far in the past literature.

5.2 Related Work

With the high popularity of blockchain, there are many survey works on research activities of blockchain and IoT systems, such as [FDM+19], [DDPS21], [DZZ19], [FS21], [PLH+21], and [HAÖG22]. Most of them focused on inventing/proposing consensus protocols, improving system performances, applications of blockchain for IoT services, and security issues. For example, as an application presented, Dhall et al. [DDPS21] provided a solution to utilize blockchain platform for reducing fake information propagation on social media/messaging systems; additionally, Hayyolalam et al. [HAÖG22] provided a comprehensive review on using edge-assisted solutions for healthcare systems based on IoT devices. Nevertheless, few of them mentioned the theoretical analysis research about blockchain systems; and even fewer had an eye on the theoretical modeling on the tangle dynamics of the IOTA blockchain network.

5.2.1 Theoretical Work in IOTA (DAG-Based) Blockchain

Though there are very limited relevant theoretical works, we observed several attempts on analytical performance modeling of DAG-based blockchain systems. Kusmierz et al. [Kus17], [KSG18b] built a rule-based discrete- and continuous-time models for IOTA, in order to build a relationship of the number of tip vertices and the vertices' cumulative weights over time. In [KG18], it theoretically analyzed the probability of being left-behind of confirmation of a message in IOTA tangle by simulating the IOTA protocol. Popov et al. [PSF19b] analyzed the message attachment behavior of the IOTA network and proved that there exists a Nash equilibrium, revealing that selfish nodes will cost more than nonselfish nodes. Our interest in this work targets to a different goal, which aims to theoretically model how the tangle topology evolves and what a degree distribution could best represent it.

5.2.2 Network Graph Models

Network graph modeling is an active research area. The famous growing/evolving network model (i.e., PA model) was proposed in [BA99]. In this model, new vertices prefer to attach on existing vertices with higher degrees, which models a common phenomenon where the rich becomes richer. The authors proved that the graph will become a scale-free network (i.e., a PL distribution) at the end.

As a variant of the PA model, CPA was introduced in [KP13]. The attachment probability of the CPA model depends on the shortest path from the node to all other nodes. The author used this model to analyze the real-world network, such as Facebook and company directors. They showed that the CPA model can provide more flexibility to model the networks in the real life. Furthermore, in [WGYZ09], another PA model's variant is proposed to model a phenomenon where a vertex acquires a new vertex depending on the density of its local area in a graph. It also shows that a PL distribution appears. The work in [HCZ+17] introduced a burst model based on the PA model. However, this burst model only extends the PA model with a random vertex mutation behavior where a new vertex randomly duplicates to multiple ones at its original point.

Recently, Pandey and Adhikari [PA17], based on the PA model, proposed a network reconstruction model for structural reconstruction of scale-free real networks. Liu et al. [LFY$^+$19] used two jointly evolving graphs, i.e., K-partite graph and generated graph, to characterize intertype and intratype interactions among nodes, respectively, and establishes the evolving process of them. Its underlying assumption is also based on the PA model where higher degree vertices are preferred when the graphs evolve. Tajeuna et al. [TBW19] modeled the community structure changes of social networks to facilitate predictions of critical events. It applied a sliding window analysis from which it developed a model that simultaneously exploits an autoregressive model and survival analysis techniques. Qiao et al. [QYB$^+$19] proposed a variant of stochastic block models in order to characterize clusters or community structures of network data with PL degree features.

In summary, although the PA model and its variants provide decent modeling for a large number of evolving networks, to our problem, IOTA tangle evolution cannot be simplified like that due to its special burst arrival mode and the vertex selection mechanism, explained before.

In reality, many phenomena do not follow the logic of a PA model. The degree distributions of their topology are also not PL/Exp distributions. For instance, the authors, respectively, showed that the file size [Mit03b], the city size [GZS10], and mobile call graphs [SMS$^+$08b] follow dPLN distributions [RJ04b]. Comparing to them, the main challenge of this work is that IOTA is a distributed network system and its network dynamics are implicit. Hence, a correct modeling with rigorous verification is needed. In fact, initial results in [GXHD20b] already realized that the PL model does not fit the empirical data of IOTA mainnet.

5.2.3 Modeling Tools

Technically, there are two main approaches used for network modeling: 1) Master Equation System (MES) and 2) SDE approaches.

The MES approach uses the Markov chain theory to derive a set of differential equations that describe the transition of the probability distribution of an interested system state [Ros95]. For example, Wing et al. [HCZ$^+$17] used this approach and presented a generalized framework to unify different evolution stages of complex networks. Its network growing strategy is similar to the PA model. The advantages of using the MES approach are its accuracy and flexibility, while its disadvantage is that modeling with MES may render the problem intractable. We will see that our problem drops into this case. This also explains why most of the existing works only covered simplified network behaviors.

The SDE approach describes a dynamic system in a probabilistic view by introducing stochastic terms in modeling [VK76]. Reed and Jorgensen [RJ04b] explained the genesis of dPLN distribution with such an approach. The advantage of using the SDE approach is its simplicity. It can help to simplify the original problem to an easier case and get a decent approximation. Its disadvantage is that sometime it may oversimplify the problems thus lose its original properties. We will see that our problem can benefit from the SDE approach, where after approximation, the original problem becomes solvable without sacrificing any key property.

5.3 IOTA Tangle Network Dynamics

5.3.1 Modeling

Our modeling consists of two components: 1) a batch attachment model and 2) a state transition model.

Batch Attachment Model:

As explained, messages in the IOTA network arrive in batches, because different nodes may independently select the same message to attach new messages to their own tangle copies. Hence, a vertex can get multiple referencing messages after consolidation. A typical random process to model this phenomenon is a multivariate Poisson process $Poi(\lambda_t,\ \lambda_m)$, where one or more messages arrive with an average rate λ_t and an average size λ_m.

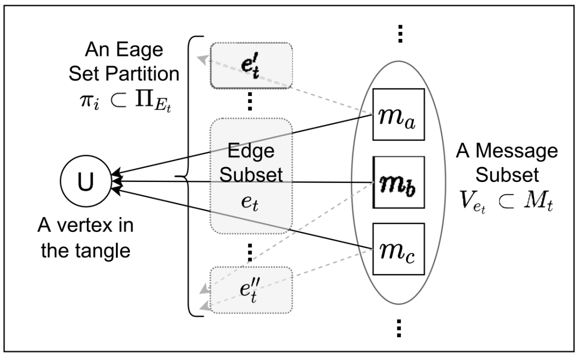


Figure 5.1: New messages and edge set partition at time t

Denoting all new messages arriving at time t as a set M_t, IOTA requires each new message (vertex) to select $s \in [2, 8]$ existing vertices in the tangle for approval. This would create maximally $s \cdot |M_t|$ new directed edges, denoted as an edge set E_t. We further denote the subset of new messages selecting the same vertex as $V_{e_t} \in M_t$; these messages will introduce a subset of new edges $e_t \subset E_t$to the selected vertex. Figure 5.1 illustrates such an example, where three new messages (ma, mb, and mc) select the same vertex U and bring three new edges to U. Note that a new message can have its new edges in multiple edge subsets at the same time (e.g., ma has its second edge to e'_t).

The total new message set M_t splits into several subsets (such as V_{e_t}), which results in a partition π_i on the whole edge set E_t split into many edge subsets (such as e'_t and e''_t). We denote all possible partitions on E_t caused by M_t's attachment as Π_{E_t} . Obviously, there are many possible ways to partition E_t, depending on where the new messages in

M_t are exactly attached/clustered. In the following analysis, it is sufficient to analyze the outcome of an edge partition $\pi_i \subset \Pi_{E_t}$, because only newly attached edges increase the vertex degree.

State Transaction Model:

The total new message set M_t always changes degrees of selected vertices and likewise, the size of the tangle. Hence, the system state of the tangle can be described with a 2-D state vector $\langle k, n \rangle$, representing a state of vertex degree type k given the current tangle size n. There are three possible state transitions involving the system state $\langle k, n \rangle$, which are illustrated in Figure 5.2 and elaborated as follows.

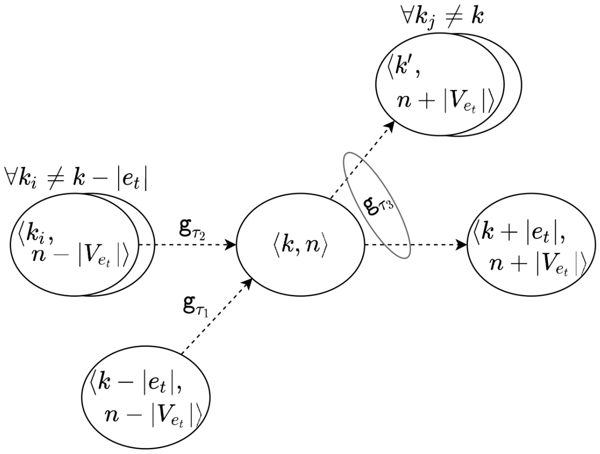


Figure 5.2: State transition graph. (For the case $k = 0$, $g_{\mathcal{T}_1}$ and $g_{\mathcal{T}_2}$ do not exist; for the case $k = K$, $g_{\mathcal{T}_3}$ does not exist.)

1) *Transition* $g_{\mathcal{T}_1}$: Suppose that the current tangle size is $n - |V_{e_t}|$, an edge subset et attaches to a type of vertex whose original degree is $k - |e_t|$. It changes the vertex's degree type to k and increases the tangle size to n, thus transiting into the state $\langle k, n \rangle$

   $\langle k - |e_t|, n - |V_{e_t}| \rangle \overset{g_{\mathcal{T}_1}}{\to} \langle k, n \rangle$

2) *Transition* $g_{\mathcal{T}_2}$: Suppose that the current tangle size is$n - |V_{e_t}|$, an edge subset et attaches to any type of vertices whose degree $\forall i = k - |e_t|$ It keeps vertices whose degree type is already k untouched while only increases the tangle size to n, thus also transiting into the state $\langle k, n \rangle$

   $\langle k, n - |V_{e_t}| \rangle \overset{g_{\mathcal{T}_2}}{\to} \langle k, n \rangle$

3) *Transition* $g_{\mathcal{T}_3}$: Suppose that the current tangle size is n, an edge subset et attaches to any type of vertices possibly with any degree. If the selected vertex has its degree type $k_j == k$, this changes the vertex's degree type to $k + |e_t|$ (i.e., the

right horizontal transition in Figure 5.2); if the selected vertex has its degree type $\forall k_j \neq k$ this changes the vertex's degree type to another $k\prime \neq k + |e_t|$ (i.e., the upper right transition in Figure 5.2). In either case, the tangle size increases to $n + |V_{e_t}|$. This makes the state jump out of the state $\langle k, n \rangle$.

$\langle k, n \rangle \overset{g_{r_3}}{\to} \langle k', n + |V_{e_t}| \rangle$

The state transition graph in Figure 5.2 is a 2-D Markov chain. The evolution of the probability distribution of the system state $p_{k,n}(t)$ follows the Chapman–Kolmogorov equation [Kar61] below:

$$\begin{aligned}\frac{dp_{k,n}(t)}{dt} = Poi(\lambda_t, \lambda_m) \cdot \sum_{\pi_i \subset \Pi_{E_t}} \sum_{e_t \subset \pi_i} (\underbrace{g_{\tau_1}(k - |e_t|) \cdot p_{\tau_1}(t)}_{Gain\ term\ 1} \\ + \underbrace{\sum_{\forall k_i \neq k} g_{\tau_2}(k_i) \cdot p_{\tau_2}(t)}_{Gain\ term\ 2} - \underbrace{\sum_{\forall k_j} g_{\tau_3}(k_j) \cdot p_{k,n}(t))}_{Lossterm}\end{aligned} \tag{5.1}$$

where each $g_{\tau_i}(\cdot)$ defines the generalized transition rate. Such a differential equation system is called a MES in statistical mechanics [Tol79], formulating the probability distribution change of a system state by aggregating all possible "Gain" and "Loss" transitions. If only the standard way of attachment is preferred, we can limit the state transitions to tip vertices with an indicator function $1(k == 0)$.

Unfortunately, the MES in Equ. 5.1 does not permit an analytical solution thus hinders our further analysis, because: 1) the MES enumerates over set partitions Π_{e_t} and further over subsets (i.e., every edge subset e_t) of every possible edge set partition $\pi_i \subset \Pi_{E_t}$. Both of them are set permutations thus do not have explicit expressions and 2) transition rate functions $g_{\tau_i}(\cdot)$ do not possess an analytical form either, as it represents a vertex selection algorithm involving subtangle operations. Clearly, a new approach is needed for the problem.

Modeling Approximation

Instead of analyzing the detailed transitions between degree types, our idea is to analyze a macro effect resulted from the new message set Mt. Recalling from Section 5.3-A, an edge set partition $\pi_i \subset \Pi_{E_t}$simultaneously leads to degree changes on multiple vertices, this motivates us to model the size change of a degree group $G_k(t)$ in a tangle, denoted as the DGS $s_k(t)$. A degree group $G_k(t)$ represents vertices all having the same degree k. In IOTA, DGS $s_k(t)$ may dynamically change due to the following two events, which are illustrated in Figure 5.3 and explained as follows.

1) *"In"-Event:* DGS $s_k(t)$ may increase, because there could be a vertex v_a, whose original degree is less than k, but an edge subset e_t makes v_a's degree increase to k with adding $|e_t|$ new edges;

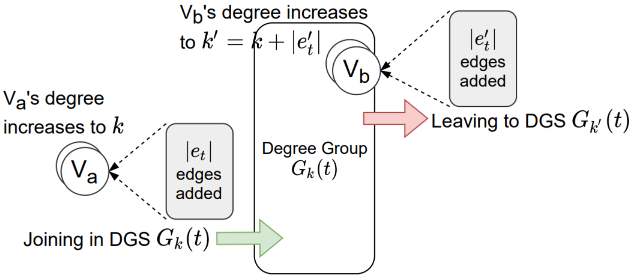

V$_b$'s degree increases to $k' = k + |e'_t|$
$|e'_t|$ edges added
V$_b$
Leaving to DGS $G_{k'}(t)$
V$_a$'s degree increases to k
V$_a$
$|e_t|$ edges added
Degree Group $G_k(t)$
Joining in DGS $G_k(t)$

Figure 5.3: Dynamics of DGS $s_k(t)$ at time t.

2) *"Out"-Event*: DGS $s_k(t)$ may decrease, because there could be a vertex v_b, whose original degree is already k, but another edge subset $e_t\prime$ makes v_b degree increase to $k\prime$ with adding $|e_t\prime|$ new edges.

In IOTA, *"In"-Event* can happen to any vertex whose degree value is between $[0, k)$, and *"Out"-Event* can happen to any vertex whose degree value is equal to k, thus covering all degree groups in a tangle.

From a probabilistic point of view, the macro effect of "In"- and "Out"-Events to a degree group G_k can be roughly viewed as a Brownian motion[LP75], because whether or not the DGS $s_k(t)$ will eventually change is uncertain, which is driven by the random vertex selections from participating nodes. Mathematically, the rate of the changing ratio can be either positive, zero, or negative during an infinitesimal period. Such a stochastic process can be formulated with an SDE of $s_k(t)$ as follows:

$$\frac{ds_k)(t)}{s_k(t)} = \omega(t)dt + \sigma dB(t) \tag{5.2}$$

where $\omega(t)$ is a growing rate coefficient, and $\sigma(t)$ is a fluctuation coefficient of random behaviors modeled as a Brownian motion $dB(t)$. Note that, we did not use the absolute change of $s_k(t)$, because the variation of $s_k(t)$ might be negative due to the Brownian motion term, which would conflict with the reality, as size cannot be negative. A benefit of using a relative ratio here is that it guarantees $s_k(t)$ a non-negative value.

5.3.2 Degree Distribution

Based on the SDE modeling, we sketch the main theoretical results regarding the stationary distribution of s_k. Since related properties are well studied, interested readers are kindly referred to [RJ04b] for the concrete steps to derive the results below.

First, the SDE in Equation 5.2 takes the form of GBM. If $\omega(t)$ and $\sigma(t)$ are independent of time t, this SDE is analytically solvable, and we have

$$s_k(t) = s_k(0) \cdot exp(\underbrace{(\omega - \frac{\sigma^2}{2})}_{\mu\ term} + \sigma B_t) \tag{5.3}$$

where the μ term is referred in the following equations.

Second, the PDF of DGS $s_k(t)$ at any observation time t follows a LN distribution:

$$f_{LN}(x) = \frac{1}{\sigma\sqrt{2\pi x}} exp(\frac{-(logx - \mu)^2}{2\sigma^2}) \tag{5.4}$$

Additionally, if the observation time t is exponentially distributed as $p_T(t) = \xi e^{-\lambda t}$, the PDF of $s_k(t)$ follows a dPLN distribution as follows:

$$\begin{aligned} p_{dPLN}(x) = \frac{\alpha\beta}{\alpha+\beta}[x^{-\alpha-1}A(\alpha)\Phi(\frac{logx - \mu - \alpha\sigma^2}{\sigma}) \\ + x^{\beta-1}A(-\beta)\Phi^c\frac{logx - \mu + \beta\sigma^2}{\sigma}], \end{aligned} \tag{5.5}$$

where $A(z) = exp(z\mu + (z^2\sigma^2/2))$, $\Phi(\cdot)$ is the CDF of a standard normal distribution, and $\Phi^c(\cdot)$ is the complementary CDF of $\Phi(\cdot)$. Although the form of dPLN distribution in Equ. 5.5 looks complicated, it can be interpreted as a multiplicative process of LN quantities over exponentially distributed observation time t.

To our problem, the interpretation is that the DGS $s_k(t)$ grows along with the tangle over time t and the stoppage time t is assumed exponentially distributed. Importantly, the PDF in Equation 5.5 tells what the probability density the size of a certain degree group $G_k(t)$ will be. After normalized with the total tangle size n, it represents exactly the degree distribution of a tangle that we target.

5.4 Parameter Estimation

5.4.1 Problem Formulation

The PDF of a dPLN distribution can be converted to a more friendly form—normal-Laplace (nLP) distribution—by substituting $y = logx$

$$\begin{aligned} f_{nLP}(y) = \frac{\alpha\beta}{\alpha+\beta}\Phi(\frac{y-\mu}{\sigma})[R(\alpha\sigma - \frac{(y-\mu)}{\sigma}) \\ + R(\beta\sigma + \frac{(y-\mu)}{\sigma})] \end{aligned} \tag{5.6}$$

where $R(\cdot) = ([1 - \Phi^c(\cdot)]/\Phi(\cdot))$ is Mills' ratio/survival function. The model parameter θ is $[\alpha, \beta, \mu, \sigma^2]$ for both dPLN and nLP distributions. In the following sections, we use the nLP distribution in Equation 5.6 for parameter estimation due to its simplicity.

Denoting the observed data (i.e., the observed degree distribution of a tangle) as $\mathcal{Y}$, the log-likelihood is written as

$$\ell_{nLP}(\theta; \mathcal{Y}) = \sum_{i=1}^{n} log f_{nLP}(\theta; y_i). \tag{5.7}$$

A corresponding MLE problem is

$$\theta^* \leftarrow arg \max_{\theta} \ell_{nLP}(\theta; y_i). \tag{5.8}$$

The problem in Equation 5.8 is usually not a concave (convex) problem due to the sum of a series of log-PDF terms. Therefore, the rest of this article focuses on the parameter estimation for the derived model, especially after we realize that in our trials generic optimization solvers cannot provide quality estimation results.

5.4.2 Main Idea

According to the result in [Ree06], the visible/observed random variable Y of an nLP distribution can be considered a sum of two invisible/latent variables Z and W (i.e., $Y = Z + W$), following Normal distribution $f_Z(\mu, \sigma^2)$ and Skewed-Laplace distribution $f_W(\alpha, \beta)$, respectively.

Based on this feature, it is possible to construct an EM algorithm [DLR77]. An EM algorithm moves to a maximized likelihood in iterations with the help of an augmented likelihood function of complete data by introducing auxiliary latent variables. Such an augmented likelihood function usually enables a simplification to the original likelihood function. Specifically, the simplified version calculates a set of expectation quantities of the augmented latent variables. This, in turn, eliminates the introduced latent variables after the expectation operation; in addition, that simplified version usually becomes a linear function of the unknown parameters, much easier for optimization.

The key benefits of an EM algorithm are: 1) neither a gradient nor Hessian matrix is needed, unlike generic optimization techniques such as Newton–Raphson methods and 2) iteration steps usually enjoy closed forms, thus quite efficient for computation. Nevertheless, a known obstacle of adopting EM framework is that no generic way exists to transform an MLE problem automatically into a form suitable in the EM framework. Always, a case-by-case design/transformation is needed, for which we will develop upon next.

5.4.3 dPLN EM Algorithm

The PDF of an nLP distribution with the visible random variable Y can be considered the marginal PDF of a joint distribution $f_{Y,Z}(\cdot)$ integrating over an introduced latent variable z. Hence, the likelihood in Equation 5.7 is extended as

$$\ell_{nLP}(\theta;\mathcal{Y}) = \sum_{i=1}^{n} \int f_{Y,Z}(y_i, z; \theta) dz \tag{5.9}$$

The following result gives a lower bound of $\ell_{nLP}(\theta, \mathcal{Y})$.

Theoream I: A lower bound $Q(\theta)$ of the likelihood $\ell_{nLP}(\theta;\mathcal{Y})$ in Equ. 5.9 is

$$\ell_{nLP}(\theta;\mathcal{Y}) \geq Q(\theta) \overset{def}{==} \sum_{i=1}^{n} E[\log f_{Y,Z}(y_i, z, \theta)] \tag{5.10}$$

where $E[\cdot]$ is the expectation over an arbitrary distribution $g(z)$ of Z.

Theorem 1 says that we can consider to maximize $Q(\cdot)$ instead of $\ell_{nLP}(\cdot)$ in Equation 5.9. The question is how to find a proper $g(z)$.

Since $g(z)$ used in $Q(\cdot)$ can be arbitrary, it is convenient to use the conditional probability $f_{Z|Y=y_i}(\cdot)$ as $g_i(z)$, which represents, how likely z will be in terms of an observed data point $y_i \in \mathcal{Y}$ with a specified parameter $\theta(s)$. This gives us an explicit $g_i(z)$ as follows:

$$\begin{aligned} g_i(z) &= f_{Z|Y=y_i}(z;\theta^{(s)}) \\ \frac{f_{Y,Z}(y_i, z; \theta^{(s)})}{f_Y(y_i;\theta^{(s)})} &= \frac{f_Z(z;\theta^{(s)}) f_W(y_i - z;\theta^{(s)})}{p_{nLP}(y_i;\theta^{(s)})}. \end{aligned} \tag{5.11}$$

Note that $g_i(z)$'s exact form in Equation 5.11 is completely given since $\theta^{(s)}$ has a specific value and all PDFs are known to us.

With $g_i(z)$, the lower bound $Q(\cdot)$ in Theorem 1 also gets an explicit form as follows:

$$\begin{aligned} Q(\theta;\theta^{(s)}) = nlog(&\frac{1}{\sqrt{2\pi\sigma^2}}) - \frac{n\mu^2}{2\tau^2} + n\log(\frac{\alpha\beta}{\alpha+\beta}) \\ &+ \frac{\mu}{\sigma^2}\sum_{i=1}^{n} E[z_i] - \frac{1}{2\sigma^2}\sum_{i=1}^{n} E[z_i^2] \\ &+ \beta\sum_{i=1}^{n} E[y_i - z]_{-\infty}^{y_i} - \alpha\sum_{i=1}^{n} E[y_i - z]_{y_i}^{+\infty} \end{aligned} \tag{5.12}$$

Let us review the two important features of $Q(\theta;\theta^{(s)})$ as follows.

1) If the four summation terms with the four $E[\cdot]$ are treated as coefficients, $Q(\theta;\theta^{(s)})$ only contains seven terms, much simpler than Equation 5.9 with n terms (i.e., the number of data points).

2) $Q(\theta;\theta^{(s)})$ is (almost) a linear function of elements $[\alpha,\beta,\mu,\sigma^2]$ of θ after the values of the four $E[\cdot]$ quantities are determined, then much easier for optimization.

These two features match our initial expectations. More importantly, these two features also provide the algorithmic procedures of our dedicated EM algorithm.

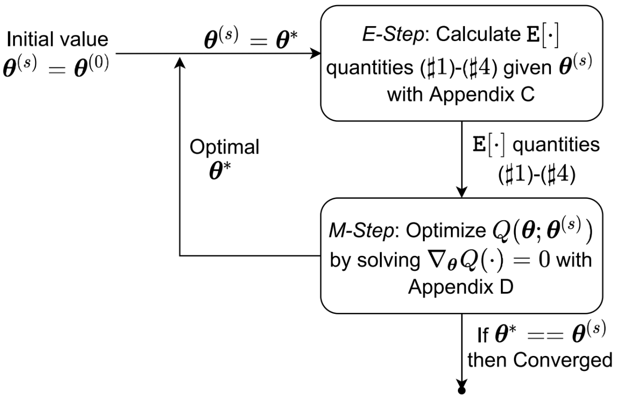

Initial value
$\boldsymbol{\theta}^{(s)} = \boldsymbol{\theta}^{(0)}$
$\boldsymbol{\theta}^{(s)} = \boldsymbol{\theta}^{*}$
E-Step: Calculate $\mathrm{E}[\cdot]$ quantities (♯1)-(♯4) given $\boldsymbol{\theta}^{(s)}$ with Appendix C
Optimal $\boldsymbol{\theta}^{*}$
$\mathrm{E}[\cdot]$ quantities (♯1)-(♯4)
M-Step: Optimize $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(s)})$ by solving $\nabla_{\boldsymbol{\theta}} Q(\cdot) = 0$ with Appendix D
If $\boldsymbol{\theta}^{*} == \boldsymbol{\theta}^{(s)}$ then Converged

Figure 5.4: dPLN model parameter estimation algorithm illustration

Feature 1 defines an "E-Step" to calculate the four $E[\cdot]$ quantities (so as the summation terms) by assigning $\theta^{(s)}$ a specific value, starting with an initial guess $\theta^{(0)}$. When calculating the four $E[\cdot]$ quantities, the introduced latent variable z is thus eliminated with expectation operations. Since summation terms become coefficients, $Q(\cdot)$ reduces to a linear form of parameter θ. $\theta^{(s)}$ will be repeatedly updated with a new value in an "M-Step" below.

Feature 2 defines the M-Step to optimize the $Q(\cdot)$ function. In this step, only the model parameter θ is treated as a variable, because the four $E[\cdot]$ quantities, which were already fixed in the E-Step, have become coefficients. To yield an optimal θ^* in this iteration, we maximize $Q(\cdot)$ by taking partial derivatives in terms of θ and solving an equation system $\bigtriangledown_\theta Q(\cdot) = 0$. As $Q(\cdot)$is linear to θ, this equation system has an analytical solution. Hence, the optimal θ^* can be directly calculated with the closed form.

Iterating between the two steps defines the body of our parameter estimation algorithm. As said, the old value $\theta^{(s)}$ is updated with θ^* derived in the M-Step, becoming $\theta^{(s+1)}$. Based on $\theta^{(s+1)}$, the E-Step recalculates the four new $E(\cdot)$ quantities, and this once again modifies $Q(\cdot)$. Then, the M-Step is repeated with the new $Q(\cdot)$ function, and it gives another θ^* used to replace $\theta^{(s+1)}$, and so on so forth. Once the optimized parameter in the M-Step does not change anymore or fulfill a predefined threshold, then a (local) optimal estimate is obtained. A diagram of the algorithm is shown in Figure 5.4.

5.4.4 Remarks

In the literature, EM-based methods are often used for estimating parameters with a mixed Gaussian model. Differently, the nLP (dPLN) model is rather a mixture of Normal and Skewed-Laplace distributions, which cannot trivially reuse the existing algorithms developed for other models. How a dPLN model can be estimated under an EM framework is partly discussed in [RJ04b] and [CFW17]. Unfortunately, none of them shows evidence of executable implementations and reports comprehensive performance evaluations; some of them even contain errors after our examination. In contrast, we not only provide explicit mathematical derivations, publish our source code but also compare its fitting performance against using existing optimization solvers.

5.5 Results

Considering the nature of the technical contributions, this work neither modified any existing system nor proposed any new system. Instead, this work theoretically analyzed the dynamics of a real-world system—IOTA network—by deriving a new model and designing a fitting algorithm. Therefore, the rationale of our evaluation plan is to directly evaluate: 1) whether the derive model (dPLN) does fit better with the observed data and 2) whether the proposed fitting algorithm provides a better parameter estimation.

5.5.1 Settings

Data Set Information

Our evaluation uses real-world historical data generated from IOTA mainnet. Some information about the used data are introduced as follows.

First, IOTA mainnet was launched on 11 July 2016. IOTA mainnet regularly maintains a network scale of more than 400 active nodes running the IOTA protocol on the Internet.

Second, the used data were officially published by IF. The whole data set contains message records from two archive periods: Period I is November 2016–June 2019 (generating 96 tangle snapshots) and Period II is April 2020–August 2020 (generating 16 tangle snapshots). Except for these two periods, we do not see any newer official data set available.

Third, tangle snapshots vary in size, which is mainly determined by the message arrival rate and the number of active participating nodes during the period of the tangle snapshot was created. The former information can be calculated by dividing the number of messages over the period length. However, for the latter information, it is difficult to restore because IF does not make the history record of participating nodes public.

Last but not least, the tangle size refers the total number of messages contained in an archived ledger snapshot. Snapshots are periodically created and archived by IF every two or three months since June 2016. After a snapshot is created, IOTA mainnet resets and starts over a new empty ledger. Hence, the tangle size is not additive between two snapshots. For a vertex's original degree, it particularly refers to its in-degree value in this work, which equals the total number of direct references from other messages.

Data Set Preparation

To prepare the reference data, given a snapshot, the in-degree of a vertex (message) can be calculated by summarizing the total number of messages that reference to the considered message. After that, for each tangle snapshot, we first count the DGS sk of every degree group G_k, and then we calculate its proportion $y_k = (s_k/n)$ in the tangle, giving the observed degree distribution of a tangle. This is the reference data $\mathcal{Y} = \{G_k, y_k\}_{k=1}^{K}$ for parameter estimation of one tangle snapshot, where each degree group G_k corresponds s_k data points (vertices).

Candidate Models

The candidate models for comparisons are listed in Table 5.1. Besides the dPLN model, the other three candidates are chosen because they are widely acknowledged network models representative for many natural phenomena. The complexity of candidate models increases from the simplest ones (i.e., PL and Exp) to complicated ones (i.e., LN and dPLN). The number of their model parameters also increases from 1 → 2 → 4. Table 5.1 also lists the solution methods of MLE for each candidate model.

Fitting Quality Metric

We use rMSLE to quantify the fitting quality. Its classic definition is given as follows:

$$rMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log y_i - \log \hat{y_i})^2} \tag{5.13}$$

where n is the total number of observed data points, and $\hat{y_i}$ is the predicted value of y_i. rMSLE can be considered a relative error of the predicted and actual values. The smallest rMSLE is zero when every predicated value$\hat{y_i}$ is equal to its observed value y_i.

The key reason to choose rMSLE is because it is a unit/scale-independent metric. Note that in our problem, the probabilities (proportions) of degree groups may differ significantly in scale. In this situation, unit-dependent measures like the Mean Absolute Error (MAE) and the Root Mean Sqaured Error (rMSE) turn out to be unsuitable, because the absolute error distances from the prediction on data points with smaller proportions will be insignificant. With those metrics, since only dominant features matter, this will falsely reflect the fitting quality. rMSLE solves this issue by taking a log-difference/relative ratio so that it becomes unit-independent.

For our evaluation, we can get a more succinct form, as the log-difference term for every data point (vertex) in one degree group G_k is identical. Therefore, we only need to calculate once the log-difference for all data points in every degree group G_k weighted by its proportion y_k as follows:

$$rMSLE = \sqrt{\sum_{i=1}^{K} y_k(\log y_k - \log \hat{y_i})^2} \tag{5.14}$$

The benefits of using Equation 5.14 are as follows:

First, it speeds up the calculation, because the summation in Equation 5.14 only has K terms (i.e., K observed degree groups), much less than the summation of n terms in Equ. 5.13. In our data sets, n means millions of vertices while K means only hundreds types of degree groups.

Second, if we remove the root and square operators in Equation 5.14, rMSLE recovers to Kullback–Leibler (KL) divergence5 that is widely used to measure the divergence between two probability distributions. Therefore, one metric acts as two. More importantly, rMSLE removes a cumbersome constraint in using KL divergence where both y_k and $\hat{y}_k$ $\forall k = 1, ..., K$ must be perfect probability distributions (i.e., the sum of probability values equal to 1). Practically, since the predicted value $\hat{y}_k$ will be sampled from a continuous PDF of a candidate model, this constraint is not always met, leading to an invalid KL divergence, thus making the evaluation fail. Then, it is inevitable to introduce extra techniques to discretize every candidate model's PDF. However, this may cause uncertainty to our evaluation, as it is unknown yet which discretization technique fits the best for our case. Instead, rMSLE measures in a similar way as KL divergence does but free of such a constraint.

5.5.2 Model Selection

Quantitative Comparison

In this part, we examine the fitting quality of the four candidate models. We rank the candidate models in a decreasing order in terms of their fitting qualities. The model getting rMSLE closest to the optimal value 0 (highlighted with yellow bar) is put at the top.

The first comparison is on overall interval where all degree groups are considered, shown in Figure 5.5. In this comparison, we show both the CDFs of rMSLEs of the four candidate models and a statistical boxplot on top. We can observe that dPLN model achieves the least average rMSLE below 0.2 with a concentrated variance distribution (shown as the short green boxplot). Besides, the LN model (in blue) ranks as the second best but its mean rMSLE (around 0.55) is already worse two times more than dPLN model's; in addition, the LN model has the largest variance of rMSLE. Furthermore, neither Expon

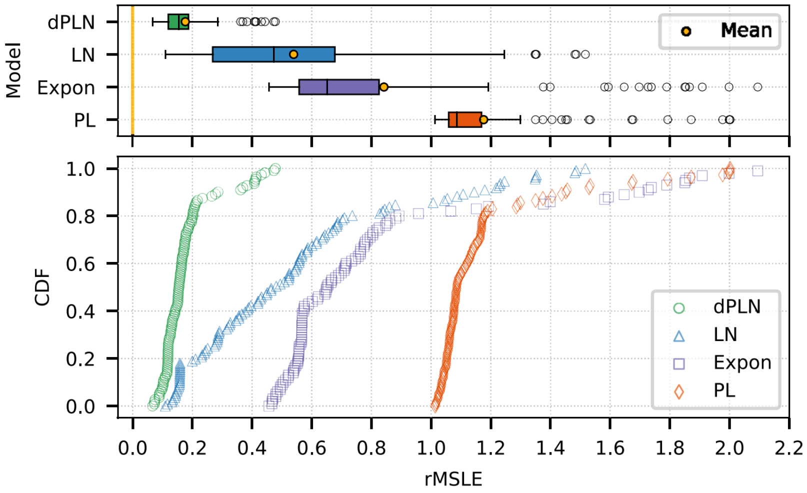


Figure 5.5: rMSLE comparison on overall interval

nor PL models (in purple and orange, respectively) seems a correct model to explain the degree distribution of the realistic tangles generated in the real world, where both of them have much worse rMSLE, especially the PL model.

The large variances of rMSLE values show the instability of the three compared candidate models when they fit the observed in-degree distributions. Actually, their mean values of rMSLE are also worse than the dPLN model's performance. In contrast, the variance of dPLN model's rMSLE is concentrated and much less than the variances of the other three models. This again justifies that the performance of the dPLN model is not only better on average but also relatively more stable than the other three models can do.

The second comparison is on segmented intervals as shown in Figure 5.6. We compare three separate intervals that split the data points into three parts: 1) header; 2) middle; and 3) rear parts. Specifically, the header part contains vertices in degree groups $G_k \in [1, 2]$, which often roughly occupy 45%; the middle part contains vertices in degree groups $G_k \in [3, 5]$, which occupy another 30%–45%; and the rear part is all the rest kinds of vertices (i.e., in degree groups $G_k \in [6, max]$).

Specifically, in the header part [shown in Figure 5.6(a)], the ranking is the same as in the overall interval but the performances of dPLN and LN models become closer, though dPLN model's rMSLE is slightly smaller. This implies that for vertices with degree values in [1, 2], both models fit well and achieve small errors. In the middle part [shown in Figure 5.6(b)], the ranking is also the same but every candidate model gets a smaller rMSLE, meaning that all candidate models fit better to the distribution of vertices with degree values between [3, 5]. Particularly, dPLN and LN models even get their rMSLE smaller than 0.1. In the rear part [shown in Figure 5.6(c)], the ranking becomes different, where the second-best model is now PL, the third place is LN, and the last one is Exp. In fact, LN and Exp show much worse performances when fitting to the degree distribution

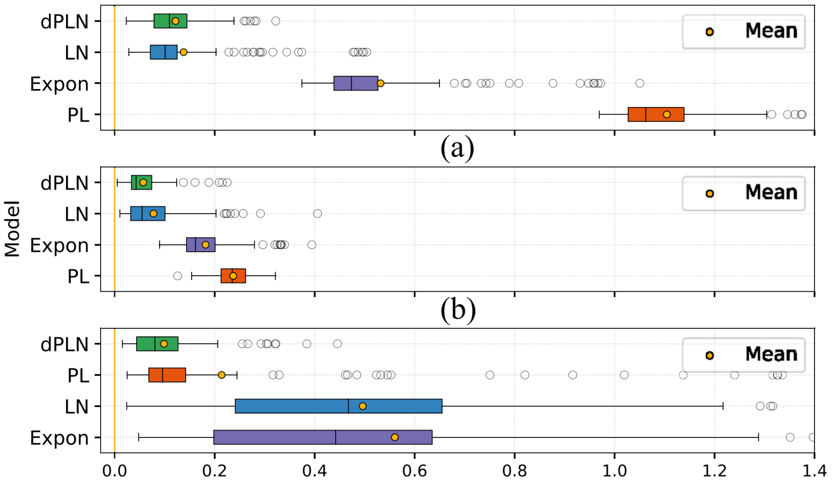


Figure 5.6: rMSLE comparison on segmented intervals

of vertices with large degree values.

Note that there are deeper reasons behind the observation where every model performs well in the middle part. We explain as follows.

1) The head part (vertices with in-degree between [1, 2]) is also a majority (45%). However, due to the shape of their distributions bending down, the other three models cannot cover the both head and middle parts. Specifically, PL model as a straight line cannot bend obviously (the worst), Exp model can slightly bend (the second worst), and LN model can do more (the third worst). Only the dPLN model can nicely balance the two parts. This is why we see distinct performances at the head part in Figure 5.6(a).

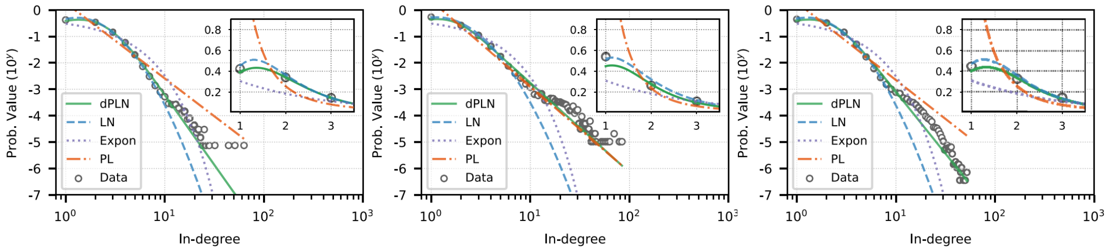
2) The rear part (vertices with in-degree ≥ 6) is not a majority ($< 5\%$). For the same reason, not all the other three models are able to cover this part. The order of thefitting performance changes, the Exp model becomes the worst, the LN model becomes the second worst, and the PL model (as a straight line to fit the right tail) becomes the third worst. Still, dPLN performs the best in Figure 5.6(c).

This observation can be seen more directly in our graphical comparison next.

In summary, we can clearly observe that on any considered interval dPLN model ranks always the best, and its rMSLEs are relatively stable.

Graphical Comparison

We then give a graphical comparison. This helps readers to understand how the four candidate models fit the reference data in a visual way. Here, in Figure 5.7 we pick three tangle instances, to which dPLN model yields its min, median, and max rMSLEs, respectively. The subplots therein particularly zoom in on the fittings of degree groups $G_k \in [1, 3]$.



(a) (b) (c)

Figure 5.7: Graphical fitting comparison of candidate models. (a) Tangle 34, (b) Tangle 39, (c) Tangle 64

In the graphical fitting comparison, we can visually observe that dPLN's fittings (the green solid curves) indeed stick much closer to the actual distribution of the data points (gray empty circles). In contrast, we can see that the other three models are far away either to the header part (such as Exp and PL models) or to the rear part (such as the LN model).

In the zoom-in subplots (on upper right corners), we can see that only dPLN and LN models can fit the data points in the header part (curves in green and blue, respectively). They are slightly different, where the LN model tends to overestimate while the dPLN model relatively underestimates the data points. Nevertheless, neither Exp nor PL model performs reasonably in the header part fitting, where the Exp model largely underestimates (purple curves) and the PL model significantly overestimates (orange curves) the data points.

The key factor making the dPLN model better than the other models is that it can not only characterize dominant features like majority vertices with degrees in $[1, 3]$ (as LN model), but also reflect special features like existences of high degree vertices (as the PL model does), which is unique to the real-world tangles. Generally, we can conclude that the dPLN model can explain much better the observed degree distributions of tangle data generated in the IOTA mainnet. This confirms our theoretical modeling for IOTA network dynamics.

5.5.3 Fitting Algorithm Comparison

We then evaluate the performance of our algorithm in Figure 5.4 named "EM" algorithm against "Broyden Fletcher Goldfarb Shanno (BFGS)" algorithm [ZBLN97b]. The BFGS algorithm is a typical example of GD-based methods already implemented in the Python.scipy package, considered a default optimization Python library. Both algorithms aim to find the optimal solution for the MLE problem defined in Equation 5.8.

We set the maximum iteration number equal to 2000 for both algorithms. We set the convergence threshold to 10^{-4} for BFGS. Particularly, we set the convergence threshold to our algorithm as follows:

$$\max\left\{\Delta_s|\Delta_s = \sqrt{(\theta^{(s+1)} - \theta^{(s)})^2}\right\} \leq 10^{-4} \tag{5.15}$$

where it requires the maximum norm of the difference in W consecutive $\theta^{(s)}$ less than 10^{-4}. Note that our threshold is harsher than BFGS uses.

For both algorithms, we evaluated ten different initial $\theta^{(0)}$ values generated with the following rules. We first fix (μ, σ^2) pair but triple the (α, β) pair from $0.1 \to 2.7$ (giving four initial values). Then, we fix (α, β) pair but triple (μ, σ^2) pair from $0.1 \to 2.7$ (giving another 3 initial values). Last, we triple both pairs $0.1 \to 2.7$ (giving the last three initial values). This gives a set of initial values differ with several magnitudes.

We did not use a random strategy to generate the initial values in order to guarantee the reproducibility of all presented results.

Algorithm Termination Status

There are three possible termination states of the two algorithm candidates as follows.

1) *"Loc-Opt.":* An algorithm terminates, because it fulfilled its convergence condition before reaching the configured maximum iteration number;

2) *"MaxIter":* An algorithm terminates, because it reached the configured maximum iteration number (i.e., 2000 here);

3) *"Boundary Condition":* An algorithm terminates, because some temporal solution violated some boundary conditions. In our case, this bound is that all elements of θ should be positive

It has to be emphasized that all three termination statuses give parameter estimation solutions but with different fitting qualities. With the ten different initial values and 112 tangle snapshots, the termination states of the 112×10 times' fitting tests with the two algorithms are summarized in Figure 5.8.

Our EM-based algorithm shows 60.36% of convergence rate, versus 15.18%, when using BFGS. In contrast, 84.82% times of using the BFGS solver triggered boundary condition versus 30.08%, when using our EM-based algorithm. Additionally, less than 10% of using our EM-based algorithm reached the maximum iteration number. Using the BFGS solver never reached the maximum iteration number, because we have seen that BFGS easily terminated due to boundary condition violation. This confirms that our algorithm has a higher chance to get a local optimal solution, which is several magnitudes higher than using the existing solver—BFGS.

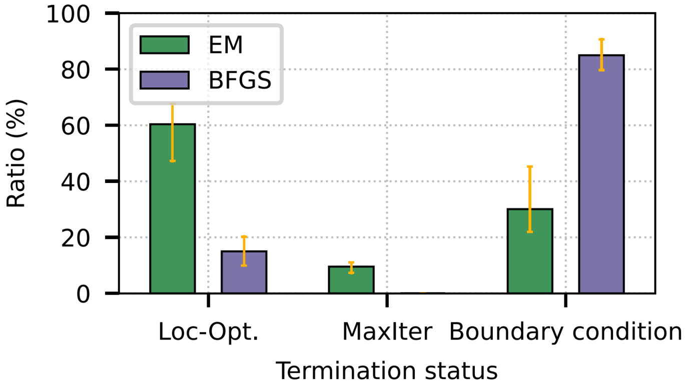
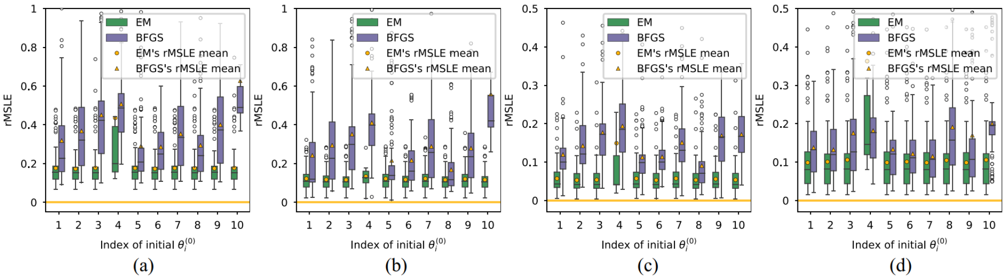


Figure 5.8: Termination status comparison.



(a)

(b)

(c)

(d)

Figure 5.9: rMSLE comparison of using EM and BFGS on (segmented) intervals.(a) Overall ($G_k \in [1, \max]$), (b) Header part($G_k \in [1, 2]$), (c) Middle part($G_k \in [3, 5]$).

Fitting Quality

We again evaluate the two algorithms with rMSLE shown in Figure 5.9. In all interval performance of using BFGS is worse than using our EM-based algorithm. We can observe a larger variation of the rMSLE of the fitting results given by the BFGS solver. Instead, the rMSLEs of our EM-based algorithm are closer to the optimal value 0 and the variations are not only more consistent but also much smaller than BFGS has. With different initial values, we observe a similar result where our EM-based algorithm achieves a better fitting quality (i.e., smaller rMSLE) than using BFGS.

The results from the termination status and fitting quality suggest that for the parameter estimation of a dPLN model, an EM-based algorithm is recommended. It also shows that it is difficult for GD-based methods to handle optimization problems within a high-dimensional space (our problem has four elements in θ thus it is 4-D). In fact, we had also tested Nelder–Mead (downhill simplex) method as a third candidate, which was chosen as an opponent that is without calculating gradient/Hessian matrix [NM65]. Since its fitting quality was even much worse than BFGS can provide, it is less valuable to report its results here.

Fitting Time

Finally, we report the time cost spent on fitting every tangle with our EM-based algorithm in Figure 5.10. The heatmap plot indicates individual execution time to reach one of the three termination states in every tangle fitting test (1120 times in total). Specifically, blocks in green, blue, and yellow colors represent termination states of "Loc-Opt.", "Boundary Cond." violation, and "MaxIter", respectively. The proportions of the three color blocks correspond to the summary result in Figure 5.8.

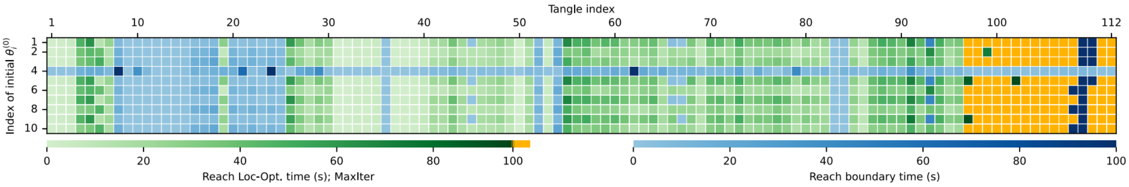


Figure 5.10: Time to reach termination statuses in all estimation tests with the proposed EM-based algorithm.

First, we observe that when estimating parameters for tangles number 8-18, 20-25, 36, 73, 74, and 109, our EM-based algorithm triggered the boundary conditions with any initial value. This is a known outcome when the given data are not perfectly dPLN distributed [RJ04b]. Second, we observe that $\theta_4^{(0)}$ seems to be a challenging initial value. With this particular initial value, our EM-based algorithm did not terminate at the Loc-Opt. status. Actually, for BFGS, with $\theta_4^{(0)}$, it also yielded poorer rMSLE. It needs further investigation to check whether such an initial value is at a location blocked to a local optimal in the solution space. Except $\theta_4^{(0)}$, our algorithm performs coherently, where we observe not only similar termination states but also similar execution times. Third, we observe that our algorithm reaches the MaxIter termination status when fitting tangles number 97–112. This can be because of our the harsh convergence threshold defined in Equation 5.15.

One important reason why the fitting time may vary among different tangles is because of the tangle size. If the tangle size n is large, the number of terms in Equation 5.9 grows as well. As a result, a fitting algorithm may take longer time in each iteration when evaluating Equation 5.9. In contrast, a different vertex's degree will not immediately influence the fitting performance. Instead, the population distribution of vertices with a certain degree will statistically influence the fitting quality of a model. The nature of the fitting data (i.e., the observed in-degree distribution) determines whether or not a model can fit well.

The execution time falls in the range with an upper limit of 100s (except for those reaching MaxIter status). From the color distribution, execution time seems more relevant to the size of the tangles (i.e., the number of vertices). Instead, it seems rather less dependent to the initial values because no matter which initial value is used, the variation of execution time across the entire data sets are similar. On average, a tangle has a million vertices, we can expect approximately 40–60s with our EM-based algorithm.

5.5.4 Statistics of Estimated Parameters

Finally, we provide a summary of the estimated parameters for the tangle data sets of IOTA mainnet, shown in Table 5.2.

As part of our future work, with the estimated parameters, the derived network model gives a new way to design an IOTA simulator. Specifically, it can initialize a dPLN distribution, then sample from the distribution, and rewire sampled vertices to construct a tangle topology. This can largely improve the efficiency because simulating heavy network protocols is avoided completely.

Model	PDF	Model Parameters	Closed-Form
PL	ζx^{γ}	γ	Closed-Form
Exp	$\xi e^{-\lambda x}$	λ	Closed-Form
LN	Equ. 5.4	μ, σ^2	Closed-Form
dPLN	Equ. 5.5	$\alpha, \beta, \mu, \sigma^2$	Algorithm 1

Table 5.1: Summary of candidate models (ζ and ξ: Normalization Constants)

Parameters	α	β	μ	σ^2
Mean	4.638	2.02	0.497	0.096
Variance	22.325	0.800	0.039	0.002
1/4 Quartile	2.076	1.526	0.432	0.101
Median	2.762	1.631	0.579	0.071
3/4 Quartile	3.766	1.881	0.624	0.131

Table 5.2: Statistics of estimated model parameters ($\theta^0 = [0.1, 0.1, 0.1, 0.1]$)

5.6 Conclusion

In this chapter, we developed a theoretical model for IOTA network dynamics with stochastic analysis. The key finding is that realistic tangles follow a dPLN distribution, which is not as usual belief, such as PL and Exp distributions. We designed a dedicated model estimation algorithm that can provide more reliable and quality solutions, which overcomes the deficiencies of using the existing solvers. Based on the real-world official data sets, the evaluation results confirmed our finding where our proposed model outperforms the existing popular network models; the evaluation results also justified the performance of our proposed parameter estimation algorithm. The whole work also gave a deeper understanding on the internal mechanisms of the IOTA network.

CHAPTER 6

Fast Tip Selection for Burst Message Arrivals on A DAG-based Blockchain Processing Node at Edge

With the rapid evolution of blockchain technology, a clear trend is that new blockchain systems (e.g., IOTA) tend to use a Directed Acyclic Graph (DAG) rather a chain structure to organize ledger records. Such a DAG-based blockchain system shows higher scalability as multiple locations are available in the ledger for new message attachment. To decide an attachment location, a popular type of tip selection algorithms follow an approach using weighted random walks on the DAG ledger. In a burst message arrival scenario, however, a processing node deployed at edge using such a method may become a bottleneck because sequentially repeating random walks significantly increases processing delay. In this chapter, we propose a new tip selection algorithm for the burst message arrival scenario on an edge node. Our solution abandons the weighted random walk approach, instead, with similar efforts we transfer to calculatein advance the tip selection probability distribution of the DAG ledger. Such a new scheme reduces tip selection to a probability distribution sampling task, which can be done extremely fast. We implement our solution and demonstrate the benefits of our approach by comparing with the random walk approach. We believe our attempt can effectively mitigate the congestion at the edge node and inspire tip selection algorithm design with a new vision for DAG-based blockchain systems.

This structure of the chapter is outlined as follows. In Section6.2, we review the existing literature; in Section 6.3, we formally introduce our Absorbing Markov Chain (AMC) modeling; in Section 6.4, we introduce our sampling-based TSA; in Section 6.5, we present our evaluation results and conclude this chapter in Section 6.6.

6.1 Introduction

A widespread adoption of blockchain technology is happening in various business sectors across FinTech [FWKKEBK20], decentralized marketplace [Sub17], and decentralized Application (dApp)s [ACA$^+$21]. A blockchain system consists of distributed processing nodes, each of which maintains a copy of a common ledger. The ledger copy on every node is synchronized and temper-proof with a distributed consensus protocol. For the first time, blockchain technology makes information shared on the Internet trustworthy in a decentralized manner.

A recent trend promotes using a DAG to organize ledger records (e.g., IOTA [Pop16]) on processing nodes, rather than a chain of transcation blocks widely used in traditional blockchains (e.g., Bitcoin [Nak09] and Ethereum [Woo14]). In a DAG ledger, a vertex represents a single message entry; and a directed edge represents an approval from the pointing vertex to the pointed vertex. Such a change brings several promising features e.g., multiple locations for new message attachment, lightweight consensus procedures, no miner/transaction fee and so on [FWS21]. All these features make DAG-based blockchain systems easier to be integrated within edge/fog computing, thus bringing IoT applications closer to the end users [WDW20].

On a processing node of a DAG-based blockchain system, a key processing logic is its tip selection module. A tip of a DAG ledger is a message without any approval. A node has to decide which tips it shall approve by attaching a new message behind there. Such a decision-making process is nontrivial because the node has to make sure that: the selected tip(s) and all their connected vertices in the branches do not conflict with the new message; in addition, hopefully, the branches of the selected tips can be re-selected with a higher chance afterwards, so that the new message itself can get approved earlier as well. The more (direct or indirect) approvals a vertex gets, the higher the weight the vertex earns. The cumulative weight is an important metric indicating how many times a message was repeatedly voted in history.

Many existing TSAs widely used on the processing nodes are designed based on the cumulative weight metric, where typically a random walk is simulated on a weighted DAG constructed from the ledger. Due to the heterogeneous weight distribution among vertices, a random walk will bias to some tips. The chance of a tip being selected reflects the collective opinion of all nodes cast in previous attachments. However, such an approach sometimes is inefficient as a processing node indistinguishably repeats random walks for tip selection of every single message. Particularly, when messages arrive on an edge node in a burst, due to a sudden long message queue and non-negligible time of doing random walks, the node might become a bottleneck in congestion, which postpones all the following phases, such as ledger consolidation and so on. Alas, this kind of burst scenarios are common in reality, e.g., crowds aggregating at a hot spot (like sport events or a public area during rush hours) to use the same service at one place. Obviously, a revisit is needed for such a scenario.

Motivated with the above concern, we reconsider the weighted random walk approach and try to seek a new solution for an edge node in this chapter. The key idea of our solution is as follows: inspired by AMC theory, we discover that the weighted DAG enjoys a nice property where the Tip Selection Probability Distribution (TSPD) can be calculated straightforwardly. We will see that pre-calculating the TSPD is worthy because, with the TSPD information, tip selections simply reduce to drawing random samples from the derived probability distribution, which not only can be reused for multiple messages but also can be done extremely fast. This completely avoids the tediously simulating random walks on the processing node. In summary, our contributions are listed below:

- We model the DAG ledger on a processing node as an AMC; we show that the stationary distribution of the modeled AMC represents a statistical outcome of sufficient random walks on the DAG, thus giving us directly the TSPD of the DAG ledger;
- Based on the TSPD property, a sampling-based TSA is proposed for handling burst message arrivals on an edge node; the proposed TSA relies on a strategy of periodically updating the TSPD along with the evolving DAG ledger;
- We implement the proposed TSA and compare with the typical weighted random walk TSA. Evaluation results demonstrate that the proposed TSA can effectively mitigate Message Attachment Delay (MAD) in the burst message arrival scenario at edge node.

To the best of our knowledge, our work is the first to ask if there could be an alternative approach replacing the weighted random walk, especially considering a DAG-based blockchain processing node at the edge.

6.2 Related Work

In this chapter, we focus on a type of DAG-based blockchain systems where a vertex in the DAG represents a single message entry and will not develop to a multi-layer DAG topology. Exemplary systems are IOTA [Pop16] and its variants, e.g., Graphchain [BCH18] and Avalanche [RYS+20]. Note that there are many other graph-based blockchain systems such as Spectre [SLZ16] and Hashgraph [Bai16]. For instance, Spectre batches messages to blocks and then organizes them as graphs. Nevertheless, those systems are of already a mixture with many extra components, thus considered out of scope in this work.

Within the interested scope, one line of research is to propose auxiliary strategies when doing weighted random walks for tip selection. For example, a TSA was proposed in G-IOTA [BGP19], where the mechanism chooses three tips, the first two are selected by

Table 6.1: Main notations

Variable	*Definition*
G_t	DAG ledger $< V, E >$ at time t
G'_t	Sub-DAG ledger $\in G_t$
n	Vertex size of the sub-DAG G'_t
v_i	A vertex (i.e., a message entry) in ledger G_t
e_{ji}	A directed edge from v_j to v_i (i.e., v_j approves v_i)
c_i	Cumulative weight (i.e., approval count) of v_i
$\tilde{V}$	Tip set in G_t, i.e., $v_i = 0$, $\forall v_i \in \tilde{V}$
ω_{ij}	Edge weight e_{ij}, defined as $\lvert c_i - c_j \rvert$
s	Required number of tip selections for a new message
v_o	Random walk starting point/head point of G'_t
v_p	Tip vertex in G_t
p_{ij}	Jumping probability on edge e_{ij} (along reverse direction)
P_t	Transition matrix of G'_t
$\tilde{\pi}^*_t$	Tip selection probability distribution over $\tilde{V}$ of G'_t
λ	Message arrival rate with a Poisson process
τ	essage window size

the weighted random walk and the third is selected from left behind tips. Later, E-IOTA introduced in [BHP20] presented a mechanism to dynamically adjust system parameters controlling the random walk simulation to reduce the number of random walks.

Another line of research is to modify the vertex (edge) weight definition so that the random walks can achieve different purposes. For example, in [WYW21], the authors proposed a new metric, called sharpness, to describe the extreme degree in a part of the DAG. Based on the new weight definition, the proposed algorithm aims to solve the splitting and fairness problem in IOTA. In [Hal21], the authors gave a novel definition of message weight and time with integrating the information from IoT devices; after that, another TSA called best tip selection method (BTSM) was proposed to enhance the resistance to malicious attacks. Similarly, in [WZ19], the authors studied a TSA optimization problem by using tree theory. They defined new labels on vertices for random walks in the DAG for tip selection and a dynamic tree will be maintained to improve the message validation efficiency.

Generally, the common ancestor of existing works is the TSA originating from IOTA [Pop16]. Existing works are still under the framework of using a weighted random walk approach. Comparing to this, differently, our goal is to look for a new approach that does not require simulating weighted random walks for a processing node at edge.

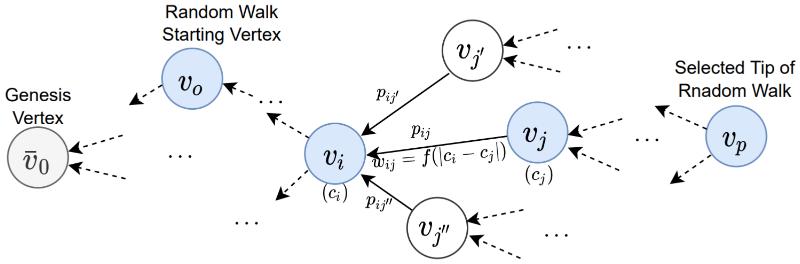


Figure 6.1: Modeling weighted random walks on DAG ledger

6.3 Modeling Random Walk on DAG As AMC

In the following discussions, we use the terms 'vertex' and 'message' interchangeably. For brevity, we also refer to 'node' as the processing node of a DAG-based blockchain system deployed at an edge network. For better readability, in addition, the main notations of this chapter are summarized in Table 6.1.

As shown in Figure 6.1, we denote a DAG ledger on a node at time t as $G_t =< V, E >_t$. A vertex $v_i \in V$ represents a message already in the ledger; a directed edge $e_{ji} \in E$ represents a direct approval from message v_j to message v_i. A directed edge e_{ji} also implies an indirect approval from v_j to all ancestor messages of v_i . G_t starts with a genesis vertex $\bar{v}_0$ (i.e., the leftmost gray vertex). The rest of the vertices can be categorized into tip/non-tip vertices in terms of their in-degree values (i.e., the direct approval count). Non-tip vertices have their direct approval count greater than 0; and tips have zero approval count (e.g., the blue vertex v_p). We further denote the set of tips in G_t as $\tilde{V}$.

We denote $v_i^{\prime}s$ cumulative weight c_i as the total count of direct and indirect approvals v_i earned at time t. This definition is also used in many existing systems such as IOTA. c_i characterizes the confidence of a message credited in the ledger. Based on c_i, we denote edge weight w_{ij} as the weight difference of its two endpoint vertices (i.e., $w_{ij} = |c_i - cj|$). The jump probability p_{ij} from v_i to any of its direct approving vertices v_j is proportional to the edge weight w_{ij}. A weighted random walk tip selection starts at a certain vertex (e.g., the blue vertex v_o in Figure 6.1), and jumps hop-by-hop with the jumping probability p_{ij} along the reverse direction of ingress edges towards a tip $v_p \in \tilde{V}$. For example, the highlighted blue vertices in Figure 6.1 form a realized path of a random walk from v_o to v_p.

Notice that tips are always the final stop of a random walk because there is no edge for further jumps. This is equivalent to the absorbing states of an AMC, i.e., a Markov chain containing states with self-transition probabilities equal to 1 (e.g., the self-transition probability of vp is 1). Actually, every random walk tip selection is a realization of state transitions on an AMC. Hence, the state transition on a sub-DAG G'_t (with a size n) starting with any v_o is fully characterized by a transition matrix P_t consisting of jump probabilities p_{ij} of all edges in G'_t including the self transition probabilities of absorbing

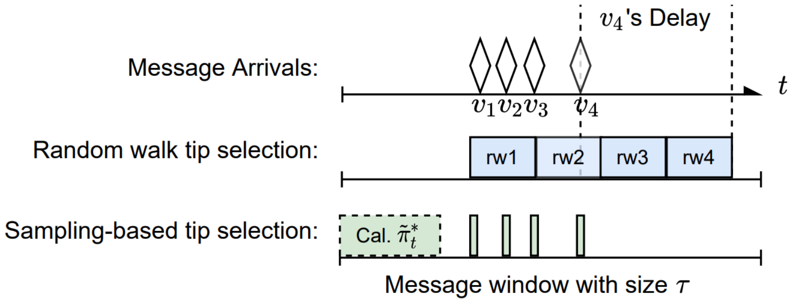

v_4's Delay
Message Arrivals:
t
$v_1 v_2 v_3$
v_4
Random walk tip selection:
rw1
rw2
rw3
rw4
Sampling-based tip selection:
Cal. $\tilde{\pi}_t^*$
Message window with size τ

Figure 6.2: Main idea illustration

states.

An important property of an AMC is its stationary distribution, denoted as $\tilde{\pi}_t^*$, which tells the staying probability of every state of an AMC in the long run. On the one hand, since non-tip vertices are transient states where a random walk never stays, the staying probability of any non-tip vertex will be zero. On the other hand, only tips in $\tilde{V}$ will yield non-zero staying probabilities. If the random walk is repeated with a sufficient number of times, the selection probability of a tip is roughly equal to the proportion of occurrences where the random walks stop at the particular tip. Thus, the statistical outcome of sufficient times of random walks is equivalent to the stationary distribution $\tilde{\pi}_t^*$ of the corresponding AMC, which just tells the TSPD of the DAG ledger at time t.

6.4 A Sampling-Based TSA For Edge Nodes

6.4.1 Main Idea

A processing node usually needs to select s tips for approval (e.g., an IOTA node picks $s \in [2, 8]$). Thus, s_k times' tip selections are needed for k messages in total. When k messages arrive at the node in a short time (i.e., a burst arrival), if the s_k times' random walks are sequentially repeated for tip selections, this may significantly increase the MAD on the node. For example, as illustrated by light blue blocks in Figure 6.2, message v_4 can be processed only if all the previous random walks are done for the first three messages (i.e., v_1 to v_3). Clearly, congestion occurs due to the close and tight arrivals on the node.

Notice that here we exclude a trivial solution: parallelizing weighted random walk simulations on the node. As explained, here we consider a processing node deployed at edge, which might have limited resources onboard, e.g., a virtualized microservice instantiated in an edge/fog computing periphery. Therefore, arbitrarily parallelizing random walks on a resource-constrained node is not an easy option.

Given such a challenge, we were wondering if the TSPD $\tilde{\pi}_t^*$ of the DAG ledger G'_t can be known in advance; if so, the node can easily sample from $\tilde{\pi}_t^*$ for tip selection without doing random walk anymore. Intuitively, such a sampling-based TSA shall be faster because: 1) the TSPD is calculated only once but reused for multiple messages; and 2) repeated random walk simulations are completely avoided, largely shortening the MAD. This idea is illustrated by the light green blocks at the lower part in Figure 6.2. The only question is: how to calculate the TSPD, which will be answered upon next.

6.4.2 Calculating TSPD $\tilde{\pi}_t^*$

The selection probability of a tip $v_p \in \tilde{V}$ equals the probability of realizing a random path leading v_o to v_p. Obviously, the random path can visit different sets of intermediate vertices (with or without overlaps) to reach the same tip v_p. For an intermediate jump, independently, it could be either a direct jump from $v_i \to v_j$ or an indirect jump $v_i \to [v_k] \to v_j$ via another vertex v_k. Assume the set of all such feasible v_k is $\hat{V}$, the jump probability of such a transition according to Champman-Kolmogorov equation [Kar61] can be formally written as:

$$\begin{aligned} p(v_i|v_j) &= p_{direct}(v_i|v_j) + \sum_{v_k \in \tilde{V}} p(v_i|v_k) \cdot p(v_k|v_j) \\ &= p_{ij} + \sum_{v_k \in \tilde{V}} p_{ik} \cdot p_{kj} \end{aligned} \tag{6.1}$$

Mathematically, Equ. 6.1 is the operation of the $i-th$ row vector multiplying the $j-th$ column vector of transition matrix P_t of G_t'. Hence, going over all rows and columns, the transition probability of one jump for all feasible cases can be calculated with a matrix product as below:

$$P^{(1)} = P_t \times P_t = P_t^2 \tag{6.2}$$

Extending the one-jump transition probability in Equ. 6.2 to multiple jumps, we have:

$$P^{(1)} = \underbrace{P_t \times \cdots \times P_t}_{\ell+1\ terms} = P_t^{\ell+1} \tag{6.3}$$

where ℓ is the path length of a random walk in G_t'. P^{ℓ} in Equ. 6.3 gives the full transition probability after ℓ jumps from *any-to-any* vertices.

In our problem, the situation is much simpler because i) the random walk always starts from a given vertex v_o(i.e., not an arbitrary vertex); and ii) the DAG ledger G_t' at any time is acyclic, thus the random walk stops in finite steps deterministically (i.e., no circle path and infinite jumps). With these nice features, we represent a random walk starting at vo as a row vector π_0(i.e., an initial state). π_0 has only the o-th element non-zero (value is 1) and its length is equal to n, i.e., the size of G'_t. Hence, with Equ. 6.3, the state transition from π_0 after ℓ jumps can be calculated by:

$$\pi^{\ell} = \pi_0 \times P_t \times \cdots \times P_t = \pi_0 \times P_t^{\ell+1} \tag{6.4}$$

The maximum value of ℓ is the maximum path length in the sub-DAG ledger G'_t, denoted by L. Immediately, this gives the stationary distribution reaching any possible tip $\forall v_p \in \tilde{V}$ in G'_t as follows:

$$\tilde{\pi}_t^* = \pi_0 \times P_t^{L+1} \tag{6.5}$$

$\tilde{\pi}_t^*$ given by Equ. 6.5 specifies the probability distribution arriving at a set of vertices after L jumps starting from a chosen point v_o. As mentioned, since the random walk only goes towards the tips, after L steps, this certainly covers the required number of jumps arriving at any other tip(s) that distance closer to v_o. Additionally, only the elements at the indices of tips (i.e., absorbing states) are non-zero in $\tilde{\pi}_t^*$, which is a known property of the stationary distribution of an AMC [Kem81].

6.4.3 The Sampling-based TSA

Knowing the TSPD $\tilde{\pi}_t^*$ facilitates a node to quickly draw any required number of random samples from the distribution for tip selection. This can handle tip selections for multiple new messages rapidly because sampling is much faster than random walk, especially useful in a burst arrival scenario. However, this approach has to consider the fact that the DAG ledger G'_t is time-evolving after adding new messages. The topology change will also alter the tip set $\tilde{V}$ so as the transition matrix Pt, thus $\tilde{\pi}_t^*$ too. In our proposed TSA, we introduce a message window size τ parameter to control the updating frequency of $\tilde{\pi}_t^*$, where after every τ s, $\tilde{\pi}_t^*$ has to be updated in order to adapt with the latest DAG ledger topology change.

Algorithm 1 Tip Selection Probability Distribution Update Worker

```
Require: $P_t$
while Timer($\tau$) is up do
    Lock $\tilde{\pi}_t^*$
    $\tilde{\pi}_t^*$ ← calculate TSPD($P_t$) with Equ. 6.5
    Unlock $\tilde{\pi}_t^*$
    Reset $\tau$ Timer
end while
```

Our sampling-based TSA mainly consists of two modules: The first module (pseudo code in Algorithm 1) is a periodic TSPD update worker at the beginning of every message

Algorithm 2 Sampling Routine

```
Require: msgQ
loop
    if msgQ.IsNotEmpty() AND $\tilde{\pi}_t^*$ is unlocked then
        $v_m \leftarrow$ msgQ.Dequeue()
        $\tilde{V}_m \leftarrow$ samplingFrom $\tilde{\pi}_t^*$ for $v_m$
        Update $P_t$ based on $G_t \cup v_m, E_{v_m,\tilde{V}_m}$
    end if
end loop
```

window. When calculating TSPD, accessing $\tilde{\pi}_t^*$ will be blocked until its updating is finished at the worker side. Locking the $\tilde{\pi}_t^*$ prevents conflict accessing from the sampling module. The second module (pseudo code shown in Algorithm 2) is the sampling routine for every new message vm suspending in the message queue msgQ, according to the derived $\tilde{\pi}_t^*$ periodically updated by the TSPD worker in Algorithm 1.

6.4.4 Remarks

First, calculating Equation 6.5 practically is not time-consuming. The reasons are: with a specified starting point v_o (i.e., π_0 vector), Equation 6.5 reduces to a vector-matrix product. This is much faster than matrix-matrix product operations, in both time and space complexities; in addition, for a DAG, the transition matrix P_t is always an upper-triangle sparse matrix, thus the actual complexity of the sparse vector-matrix products in Equation 6.5 is much lower than normal dense matrix multiplications. In our evaluation, its cost is slightly more than a single-time random walk simulation.

Second, although calculating a TSPD consumes slightly more time, such overheads pay off because the TSPD information can benefit to the tip selections of following messages dropping into the same message window while the weighted random walk cannot. The delay can be largely mitigated after knowing the TSPD because a rapid sampling from the TSPD $\tilde{\pi}_t^*$ replaces the random walk for every message (as illustrated by the blue blocks in Figure 6.2).

Third, the proposed TSA is backward compatible because it is an optimization to the local tip selection module on one node, which only relies on the information already available from the node and does not require any external interaction with neighboring nodes. It is not mandatory to have a full installation of our TSA to the whole DAG-based blockchain system.

Last but not least, the proposed TSA does not touch the principle of tip selection. An invalid message is treated in the same way, i.e., drop and re-sample a tip until a valid one is identified. In other words, the proposed TSA considers the efficiency issue when selecting a tip, thus neutral to all following stages.

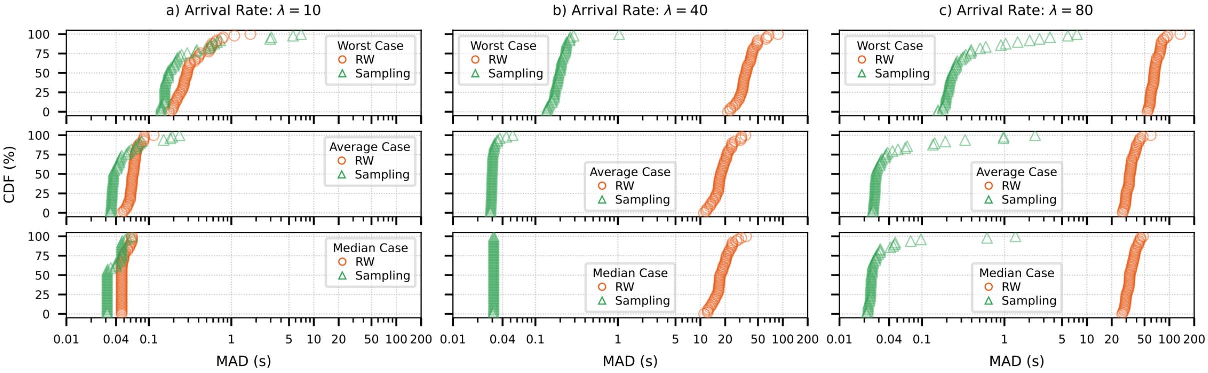


Figure 6.3: Performance comparisons on Message Attachment Delay (MAD) δt_i ($\tau = 0.06s$)

6.5 Evaluation Results

We implemented our sampling-based TSA (labeled as ‘Sampling’) in Python and compared with the typical weighted random walk TSA (labeled as ‘RW’). In all evaluation tests, we set our sub-DAG G'_t size $n = 1000$, and we set $s = 3$ (i.e., three tips have to be selected for each message).

6.5.1 Message Attachment Delay (MAD)

Given three different scales of the message arrival rate λ, we first evaluated the two methods with MAD defined as $\delta t_i = t_i^c - t_i^a$, where t_i^a is the time a message enters the message queue and t_i^c is the time its three tips are selected. For each λ, we repeated the test $K = 50$ times and in each time we randomly generated $m = 2000$ messages. We are concerned with the median, mean and worst cases of MAD with the two methods. The results are shown in Figure 6.3.

When the arrival rate is low ($\lambda = 10$), the two methods have similar performances as shown in Figure 6.3a’s column. For the median MAD, both methods could make half of the traffic loads experience MAD around $0.04s$, where the Sampling method performed slightly better; for the worst case, few more numbers of tests with our method showed longer MAD (prolonging to the $[1s, 10s]$ interval), this also worsened the mean MAD of some tests with our method. As expected, the proposed Sampling method does not enormously advantage in a low arrival rate scenario due to similar costs of updating TSPD $\hat{\pi}_t^*$ once and doing a single random walk.

However, when the arrival rate increases (e.g., $\lambda = 40$), as shown in Figure 6.3b’s column, the performance of the RW method severely degraded, where half of the traffic loads (i.e., the median case) experienced their MADs in between 10 s and $50s$; for the worst case, there were 20% of tests experiencing MAD $> 50s$. Instead, the proposed Sampling method did not degrade. Clearly, with a burst arrival, more messages dropped in the same message window, and thereby could reuse the calculated TSPD $\hat{\pi}_t^*$ for tip selection

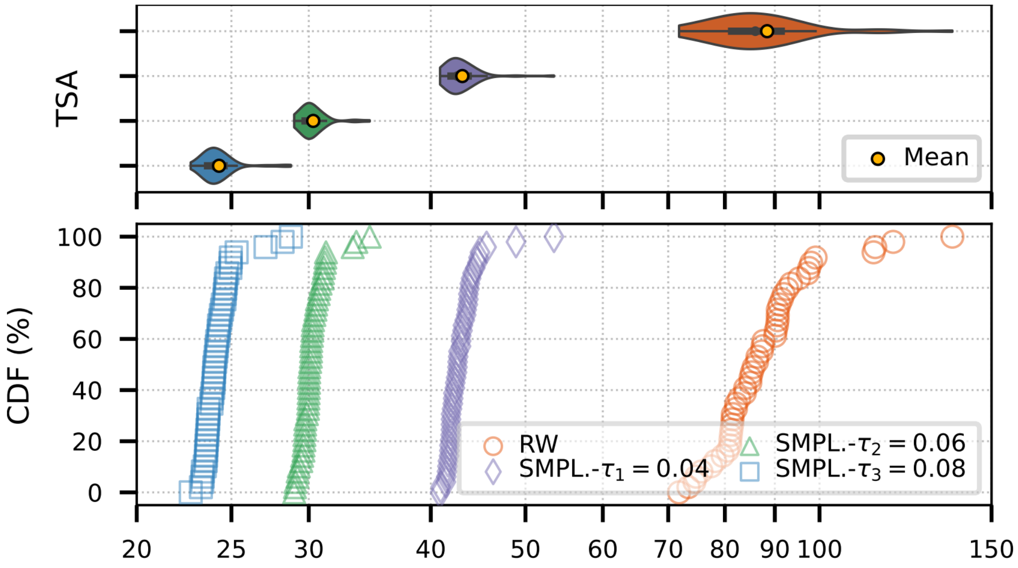


Figure 6.4: Comparison on total processing time ($\lambda = 40$)

by sampling. Similarly, when the arrival rate doubled to $\lambda = 80$, as shown in Figure 6.3c's column, the median and mean cases of MAD with the RW method prolonged to the interval of $[20s, 50s]$ and its worst case even prolonged beyond $100s$ in some tests. Instead, our method showed that only less than 20% of tests in the worst case prolonged beyond $1s$ but still less than $10s$.

The MAD evaluations clearly confirmed our motivation and the key benefit of the proposed sampling-based TSA, where the processing delay at the node can be effectively mitigated especially in a burst message arrival scenario.

6.5.2 Total Processing Time

We then evaluated the total processing time T consumed by the two methods for the tip selections for all new messages, given three different message window size τ values $0.02s$, $0.04s$ and $0.06s$. Similarly, for each τ value, we repeated the tests $K = 50$ times and each test processed $m = 2000$ messages but with a fixed arrival rate $\lambda = 40$. The evaluation result is shown in Figure 6.4.

First, our proposed TSA consumed much less time to finish tip selection for all messages than the RW method did (see the three cold-color curves are all at the left-hand side of the orange curve). Specifically, with our method, nearly 90% of the tests consumed around $43s$, $30s$ and $23s$, respectively to finish the entire jobs. In contrast, 80% of tests with RW method consumed $70s$ to $90s$ and the rest took up to $150s$. This confirms that with the message window size τ increasing, the frequency of updating the TSPD of every message window became less often, while the chances of reusing a derived TSPD increased gradually. This also shows the influence of the message window size τ.

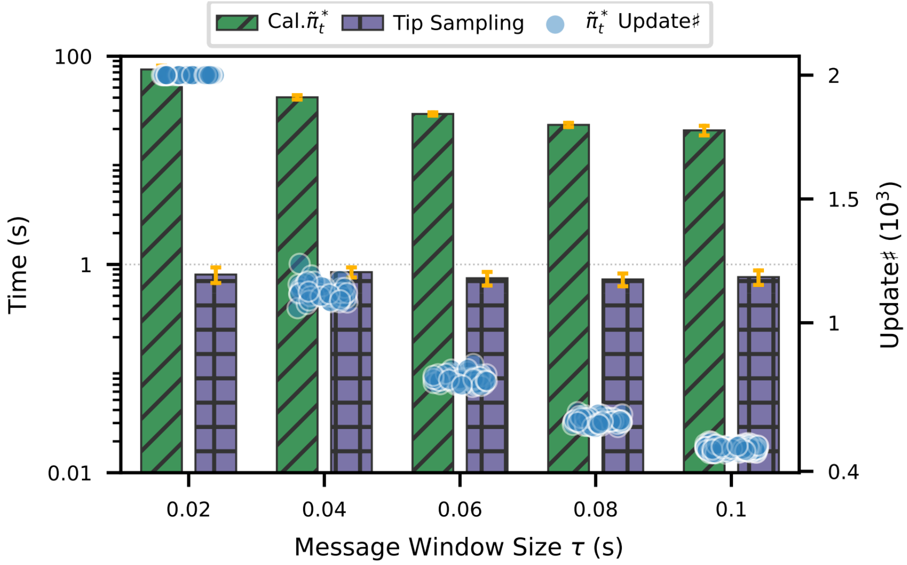


Figure 6.5: Time ratio of the two modules in Sampling method ($\lambda = 40$)

The evaluation results of the two different metrics above confirm the idea of the proposed Sampling method, which says that paying slightly more efforts to calculateTSPD $\hat{\pi}_t^*$ is definitely beneficial. They also confirm that simulating weighted random walk is not imperative to tip selection in DAG-based blockchain systems. Instead, we do have a better approach to achieve the same goal instead.

6.5.3 Features of the Proposed TSA

Last, we are also interested in the features of the proposed Sampling method. Our Sampling method pays main efforts on periodically updating a TSPD $\hat{\pi}_t^*$, which is different to the RW method where all time for tip selection is mainly spent on random walks. Therefore, it is helpful to quantitatively measure the time proportion of the two modules (i.e., Algorithm 1 and Algorithm 2). The result is shown in Figure 6.5 still with an increasing τ value.

We can first notice that the proportion of time spent on calculating TSPD $\hat{\pi}_t^*$ indeed dominates, comparing with the time for sampling tips (i.e., the forward slash bars are much higher than the purple bars). This again reflects the key idea of the proposed TSA, where if the TSPD can be known, the tip selection is easier. Secondly, we can find out that when the message window size τ increases, the number of times updating the TSPD $\hat{\pi}_t^*$ (i.e., the blue point clouds) decreases from 2000 (i.e., no reuse at all) to around 450 times. This also matches our expectation where the larger the message window size τ, the less frequent TSPD updates will be.

As an initial attempt, clearly, many other interesting aspects are not covered in this work, such as impacts to the DAG topology evolution and so on, which is being undertaken as ongoing work.

6.6 Conclusion

In this chapter, we focused on the tip selection module of a DAG-based blockchain processing node. Instead of following the existing approach using weighted random walks, we proposed a different strategy that pre-calculates the TSPD of the DAG ledger then sampling for tip selection. Evaluation results confirm that the proposed TSA can largely mitigate MAD at the edge node facing burst arrivals. We believe that our new approach can further.

CHAPTER 7

A Stable and Secure Transaction Tip Selection Algorithm For IOTA

With the advent of the IoT, an amount of data is generated and processed. Ensuring the privacy, security, and interoperability of data is a challenging task. Blockchain has emerged as a potential solution to address these issues, as it is a decentralized system that ensures the integrity and confidentiality of data through encryption. However, the scalability of blockchain technology is still a critical limitation with the increasing volume of data. To address this limitation, DAG data structure has been proposed to improve scalability by supporting asynchronous process of transactions. IOTA is a well-known DAG-based blockchain that theoretically offers faster confirmation speeds with an increasing number of transactions. However, in practice, IOTA still faces the challenge of balancing scalability and security. In this chapter, we propose a scalable and secure transaction attachment algorithm for the DAG-based blockchain IOTA. We determine two critical parameters through empirical analysis: one for calculating the selection probability and the other for setting the threshold for abnormal transactions. Firstly, we calculate the selection probability of unconfirmed transactions. Then, we select abnormal transactions whose selection probability falls below the predefined threshold to maintain security. Finally, new transactions attach randomly to former transactions with a computational complexity $O(n)$, ensuring scalability. Through experiments comparing the proposed algorithm to the current transaction attaching algorithm, we demonstrate the scalability and security of our proposed algorithm.

The rest of the chapter is organized as follows: We illustrate the related analysis about the TSA and attacks in Section 7.2. We describe the design of the proposed algorithm in Section 7.3. In Section 7.4, we design the experiment to determine the critical parameters and test the proposed algorithm. Then we analyze the experiment result in Section 7.5. The future work is illustrated in Section 7.6. We conclude the whole chapter in Section 7.7.

7.1 Introduction

In recent years, the IoT has experienced significant growth, with an increasing number of intelligent devices. These devices generate vast amounts of data, which has led to huge challenges in data interoperability, security, and privacy [ZGK+22]. Blockchain technology as a novel DLT that provides transparency, anti-tampering, and traceability of data. Meanwhile, the blockchain system is decentralized and against the single point of failure, making it an ideal solution for the challenges faced by IIoT [ZXE+23, YW21, WCL23].

As the IoT continues to expand, the number of transactions submitted to the blockchain is increasing at an unprecedented rate. However, this rapid growth has led to a shortage of blockchain resources, with scalability being one of the most critical issues. In comparison to centralized systems, the confirmation speed of blockchain is relatively slow, and the TPSof traditional blockchain technology is insufficient to meet the high throughput requirements of IoT use cases [KKKR22].

(a) Chain

(b) DAG

Figure 7.1: A comparison of the blockchain data structure

A novel blockchain data structure DAG is proposed to solve the scalability issue. As shown in Figure 7.1, compared to the chain, blockchain with DAG can process transactions asynchronously. The vertex on the DAG represents either a block of transactions or a single transaction. The directed edges of the DAG indicate the confirmation relationship. In these years, various DAG blockchains have been developed, such as IOTA[Pop16], Byteball [Chu16], Hashgraph [BL20] and Fantom [NCK+21] etc. IOTA is one of the most widely deployed DAG DLTs, which is maintained by IOTA Foundation (IF) [1].

There exist two versions of IOTA, namely IOTA 1.0 and IOTA 2.0, with the latter being the most recent. They differ in their consensus mechanisms [Pop16] [PMC+20]. Despite the novelty of IOTA 2.0, IOTA 1.0 is being still used in both research interest [TCS+22, RID+23] and applications [ASA+23, ZZSS22, EJCF22, PSV+22, MWB+22] recently due to its specific consensus mechanism. Therefore, IOTA 1.0 still holds potential for development and improvement. The term "IOTA" in the following context refers to IOTA 1.0.

In IOTA, the DAG data structure is referred to as the tangle, where each vertex represents a transaction. Upon the arrival of a new transaction, it must select and approve two previous unconfirmed transactions, which are also called tips. The algorithm used for selecting tips is named TSA. The original IOTA protocol employs the MCMC algorithm as its TSA, which utilizes a weighted random walk to attach new transactions. A critical

[1] https://www.iota.org/

parameter, denoted as α, is used in the MCMC algorithm. A larger value of α results in a more secure IOTA, as the walker is more likely to traverse transactions with higher weights. However, this also means that transactions with lower weights, even if they originate from honest nodes, may not be approved. Consequently, a larger α value leads to an increase in unconfirmed transactions. Conversely, a smaller α value may reduce the number of unconfirmed transactions but increase the selection probability of abnormal transactions, which are attached to the tangle not through MCMC. To enhance the IOTA's defense against attacks, a larger α value must be set, which will result in more unconfirmed transactions in the tangle. Therefore, IOTA with the MCMC algorithm still struggles to balance security and scalability.

There have been several research efforts aimed at stabilizing and minimizing the number of unconfirmed transactions, meanwhile keeping the security of the tangle. One such effort was proposed by G. Bu et al. in the form of G-IOTA [BHP20]. This approach involves each new transaction referencing three previous messages. The same team later proposed E-IOTA [BGP19], a variant of IOTA that utilizes a mix of TSA with varying α values executed with different probabilities. For each round, one of three α values is used to perform a random walk and select the tip. In DA-IOTA [RID+23], S. Rochman et al. set the α value as a variable that depends on the standard deviation of all cumulative approver weights. These research works have successfully controlled and stabilized the number of tips. However, the tangle remains vulnerable to attacks when a small α value is deployed. As long as a small α value is used, the tangle will remain at risk of being attacked.

Our aim is to enhance scalability while maintaining security. Initially we detect and select out abnormal tips with abnormal selection probabilities, then attach new transactions using URTS, which selects the tip from set of all tips randomly [KSP+19b]. There are two main challenges to achieve the goal:

1. *A Proper α for the Tip Selection Probability Calculation:* The parameter α directly influences the probability of selecting tips in a tangle when the new transactions are attached via MCMC. In such a tangle, tips on the random walk routine with higher weight may have a greater selection probability. However, the situation may differ in a tangle generated via URTS. Therefore, selecting an appropriate α that is sensitive to abnormal tips and attack patterns is the first challenge of this study.
2. *A Baseline Value for the Abnormal Tip Selection:* In order to identify abnormal tips, a baseline between the selection probabilities of normal tips and abnormal tips needs to be established. This baseline may vary depending on the transaction incoming rate λ and the weighted random walk parameter α. The accuracy of tip detection is also influenced by the baseline. Therefore, determining an appropriate baseline represents the second challenge.

This chapter proposes a Secure Uniform Random Tip Selection (S-URTS) algorithm that addresses the aforementioned challenges and ensures the scalability and security of the tangle. Our solution effectively mitigates the risk of attacks by detecting them prior to attaching new transactions, thereby maintaining a stable number of unapproved transactions. The previously attached transactions can be approved immediately by the incoming transactions, and the new transactions will be approved in the subsequent round, without any accumulation of unapproved transactions. Our contributions are as follows:

- We propose a novel tip selection algorithm, which can maintain both scalability and security of a DAG-based blockchain.
- We determine a proper α for the proposed algorithm based on an empirical analysis.
- We set the baseline for the normal tip distribution and detection of the abnormal tips.
- We demonstrate the properties of the proposed algorithm through various experiments. The proposed TSA-URTS takes similar time to other TSAs, but S-URTS has less number of tips and could defend against to the parasite chain attack.

7.2 Related Work

In this section, we present previous works pertaining to the scalability and security of the IOTA tangle. These works encompass theoretical analyses of tips count, tangle TSA variants, and tangle security.

7.2.1 Theoretical Analysis of the Tangle Tips Count

The experimental analysis of the influence of α and λ on the number of tips has been conducted and reported in [KSG18a]. The results of the experiment indicate that a small value of α leads to a slower development trend of tips, while a large value of α causes a continuous increase in the number of tips. Among the various TSAs, URTS exhibits the smallest number of tips, whereas MCMC has a higher number of tips than URTS, even when α is 0. This finding has also been confirmed in [KSP^{+}19a]. In another study by the same team, reported in [KG18], the influence of α and λ on the probability of left-behind transactions and permanent tips has been analyzed. The results indicate that, for the same value of λ, an increase in α leads to a higher percentage of tips.

7.2.2 Tangle TSA Variants

There exist several works proposing various algorithms to stabilize the number of tips. In G-IOTA [BGP19], the number of tips is reduced by approving three tips through a new transaction, and experimental results demonstrate a decrease in the number of tips. To reduce the number of random walks and save energy consumption, the same team proposed E-IOTA [BHP20]. For each random walk process, one α is selected from the

α set with a certain probability p. The security is maintained by a large α, while the number of tips is stabilized by another small α and 0. Experimental results confirm that E-IOTA can maintain a low number of tips. However, the security experiment is still missing, and the determination of the selection probability p is not provided. Pietro et al. proposed a hybrid TSA [FKS20] by using a large and a small α for two tip selection processes separately. It is experimentally proven that this method can stabilize the number of tips. But there is no information on how to set the two α values. A TSA algorithm DA-IOTA was proposed in [RID+23], which determines the α size based on the standard deviation of the cumulative weight. Comparing with MCMC and E-IOTA, the number of tips is smaller than the other two TSAs. However, there is no detailed explanation of the algorithm's basis and no proof of security.

All the above TSA variants have better performance than MCMC in maintaining a stable and minimum number of tips, but they lack sufficient experiments to approve their security and enough information about parameter setting.

7.2.3 Tangle Security

The most prevalent form of attack in the IOTA network is the parasite chain attack, and several studies have been conducted on detecting such attacks. One approach involves using a sampling random path to calculate a distance and identify the parasite chain, as described in [PKC+20a]. If the calculated distance d exceeds a predetermined threshold, a flag is raised, and the tip selection process needs to be restarted. Experimental results have confirmed the effectiveness of this detection algorithm. Another study by Ghaffaripour et al. [GM22a] proposes a scoring function to measure the importance of transactions in the IOTA network. Any sudden changes in transaction importance indicate abnormal behavior, which can be used to detect parasite chain attacks. Chen et al. [CGWB22] analyzed the behavior strategies of IOTA nodes using evolutionary game theory and identified key factors affecting parasite chain attacks. They proposed a parasite chain attack prevention algorithm based on price splitting, which effectively prevents the formation of parasite chains. Numerical simulations confirmed the effectiveness of the proposed solution.

While these above TSA variants and parasite chain detection algorithms have shown promising results, there is still lack of a work verifiying and evaluating both scalability and security of the novel TSA comprehensively.

7.3 Algorithm Design

This section presents the proposed TSA S-URST. Before deploying the algorithm, we need to determine two important parameters: α and T. These two parameters will influence the precision of the abnormal structure detection. To facilitate understanding, we provide a summary of the definitions of all variables used in this study in Table 7.1.

7.3.1 Determine the α

Algorithm 3 α Determination

```
Require: $set(\lambda)$, $n$, $m$, $set(\alpha)$
Ensure: $\alpha$
for $\lambda$ in $set(\lambda)$ do
    $G_\lambda = tangle\_generator(\lambda, n)$
end for
for i in [1,$m$] do
    $Pc_i = parasiteChain\_generator$(i)
end for
for $G_\lambda$ in set($G$) do
    for $Pc_m$ in set($Pc$) do
        $G_\lambda^m = parasiteChain\_attach(Pc_m, G_\lambda)$
    end for
end for
for $G_\lambda^m$ in $set(G_\lambda^m)$ do
    for $\alpha$ in $set(\alpha)$ do
        $D_\alpha^{\lambda,m} = probability\_calculator(G_\lambda^m, \alpha)$
    end for
end for
for $D_\alpha^{\lambda,m}$ in set($D_\alpha^{\lambda,m}$) do
    $p_{min}, p_{pc} = select\_from(D_\alpha^{\lambda,m})$
    $p_{diff} = p_{min} - p_{pc}$
end for
Calculate the mean and variance of $p_{diff}$ for each $\alpha$
Choose the $\alpha$, whose mean is max and var is min.
return $\alpha$
```

The value of α will have a direct impact on the probability of tip selection. As the tangle is generated through the use of URTS TSA, the effect of α on the probability of tip selection may differ from that of the tangle generated through MCMC. It is imperative that we select an appropriate value for α that can differentiate between the selection probabilities of normal and abnormal tips. In this chapter, we employ an empirical approach to determine the appropriate value for α.

Algorithm 3 shows the whole process for α determination. Firstly, we generate tangles for various values of λ using the URTS algorithm, and add parasite chains of varying lengths to the tangle. Subsequently, for each length of the parasite chain, the selection probability of both normal tips at the main tangle and the abnormal tips at the parasite chain are calculated and collected. Finally, the difference between the minimum selection probability of normal tips and the selection probability of abnormal tips is calculated. The mean and variance of these differences are then computed, and the value of α with the largest mean and smallest variance is selected.

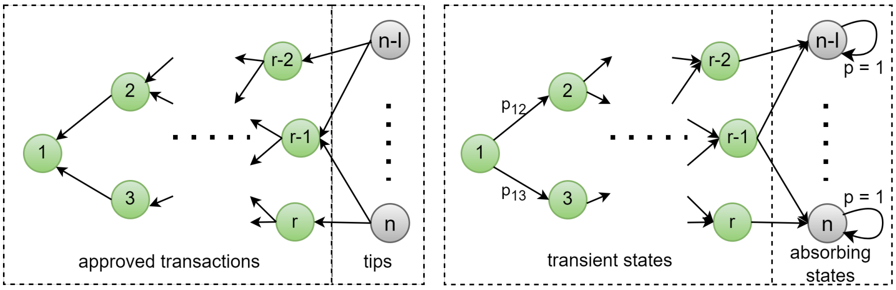
7.3.2 Determine the T

Algorithm 4 Threshold Determination

```
Require: $set(\lambda)$, $n$, $\alpha$
Ensure: $T$
for $\lambda$ in $set(\lambda)$ do
    $G_\lambda = tangle\_generator(\lambda, n)$
end for
for $G_\lambda$ in $set(G_\lambda)$ do
    $D_\lambda = probability\_calculator(G_\lambda, \alpha)$
end for
for $D_\lambda$ in set$(D_\lambda)$ do
    $D_{min} = \min(D_m)$
end for
Calculate the moving average: $T = moving\_ave(D_{min})$
return $T$
```

After determining an appropriate value for α, it becomes necessary to identify a suitable threshold T for detecting the selection probability of abnormal tips for various values of λ, shown in Algorithm 4. The moving average algorithm is utilized to determine the threshold T. Initially, we collect the values of D_t for each t during the tangle generation process. Subsequently, we obtain the minimum value of each D_t and calculate the moving average value. Once the moving average value stabilizes and converges, we set that value as the threshold T.

7.3.3 Proposed TSA S-URTS



(a) Tangle (b) Absorbing Markov chain

Figure 7.2: Convert tangle to a absorbing Markov chain

The present algorithm S-URTS commences by transforming the tangle into an absorbing Markov chain, followed egin by the computation of the probability distribution of all tips. Subsequently, the identification of the anomalous tip is carried out, and transactions

Table 7.1: Variable definition

Variable	*Definition*
G_t	The DAG at time t
n	The number of transaction
Pc_i	The parasite chain i
m	The length of the parasite chain
λ	The new transaction arrival rate
E_t	The edge set at time t
e_{ij}	The edge between two adjacent messages i and j
V_t	The transaction set at time t
v_i	The transaction i of the tangle
v_0	The genesis transaction of the tangle
L_t	The set of tips at the time t
l_t	The number of tips at the time t
D_t	The probability distribution of tips at the time t
P_t	The transition probability matrix at time t
p_{ij}	The transition probability between message i and j
π_t	The probability distribution of the absorbing state at time t
α	The weighted random walk parameter
c_i	The cumulative weight of message i
w_{ij}	The edge weight of edge ij
$T(t)$	The tip selection threshold at the time t

are selected from the remaining tips. The primary steps involved in the algorithm are illustrated in Algorithm 5.

At first, we transform the tangle G_t into an absorbing Markov chain via designating tips as absorbing states and reversing the direction of directed edges in the tangle. For example, the tangle shown in Figure 7.2a includes n transactions, comprising r approved transactions and l tips. Figure 7.2b shows the absorbing Markov chain converted from that tangle in Figure 7.2a, which includes r transient states and l absorbing states with a transient probability of 1. The transient probability from state 1 to states 2 and 3 is p_{12} and p_{13}, respectively.

Then, we calculate the cumulative weight c_i of each transaction i and get the edge weight w_{ij} of each edge ij from Equation 7.1. The affinity value between two states a_{ij} is influenced by α and calculate by Equation 7.2. We obtain the transition probability p_{ij} for each pair of connected transactions from Equation 7.3. After gathering this information, we construct the transition matrix P of the absorbing Markov chain, initiate the initial state π_0 as Equation 7.4, calculate the stationary state distribution π to obtain the tip selection probability distribution D_t, through Equation 7.5.

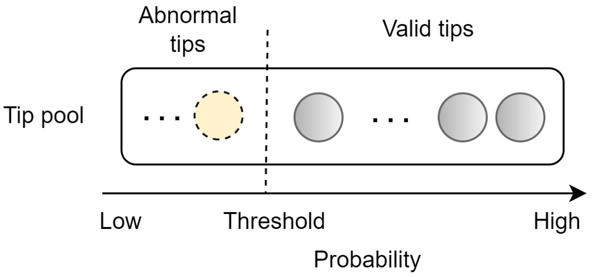


Figure 7.3: Illurstration of the abnormal tips selection

$$w_{ij} = c_i - c_j \tag{7.1}$$

$$a_{ij} = exp(-\alpha w_{ij}) \tag{7.2}$$

$$p_{ij} = \begin{cases} a_{ij} / \sum_{z \in N(i)} a_{iz}, & 1 \leq i \leq r, \\ 1, & i = j,\ n - l \leq i \leq n, \\ 0, & \text{otherwise}. \end{cases} \tag{7.3}$$

$$\pi_0 = [1, 0, ..., 0] \tag{7.4}$$

$$\begin{aligned} &\pi_1 = \pi_0 P \\ &... \\ &\pi = \pi_0 P^k \end{aligned} \tag{7.5}$$

At the end, we pick out the abnormal tips as shown in Figure 7.3. We select the tips whose selection probabilities are below the threshold $T(t)$, and delete these abnormal tips from the tip set, construct a new tip set $L'(t)$, and attach new transactions to the new tip selecting from set $L'(t)$ uniformly.

In order to improve the efficiency and energy utilization of adding new transactions, and to avoid network congestion, new transactions are added at a fixed time unit interval. The current set of newly arrived transactions is $M(t)$ and the new transactions are m_1,m_2,... . The above process is executed once for every time unit, and the new transactions are added to the new tip set $L'(t)$ in the order they arrive. This process ensures that the new transactions are added to the tip set in a timely manner, and that the network does not become too busy.

As the DAG topology evolves over time with the addition of new transactions, the proposed TSA S-URST algorithm effectively adapts to these changes by employing a probabilistic approach within a fixed time window. The algorithm recalculates the selection probabilities of unconfirmed transactions based on a pre-set random walk weight parameter. This recalibration allows the algorithm to dynamically account for changes in the tangle's structure caused by the continuous arrival of new transactions. To maintain the integrity of the tangle, the algorithm incorporates an anomaly detection and pruning mechanism, utilizing pre-defined thresholds to identify and exclude anomalous transactions. This ensures that irregularities in the DAG topology are effectively managed. By continuously recalculating probabilities and pruning anomalies, the framework adapts seamlessly to the evolving tangle structure, guaranteeing robust and secure operation in dynamic environments.

Algorithm 5 Tip Selection

```
Require: $G(t)$, $V(t)$, $E(t)$, $\alpha$, $\lambda$, $T$
Ensure: $tip_1$, $tip_2$
for $v_i$ in $V(t)$ do
    $c_i = sum(children(v_i)) + 1$
    if in-degree($v_i$) $= 0$ then
        Add $v(i)$ to the $L(t)$
    end if
end for
for $e_{ij}$ in $E(t)$ do
    $w_{ij} = c_i$ - $c_i$
end for
for $e_{ij}$ in $E(t)$ do
    $p_{ij} = f(e_{ij}, \alpha)/sum(f(e_{ij'}, \alpha))$ for all $j' - > i$
end for
Construct the transition probability matrix $P_t$
Calculate the stationary state $D(t)$
for $d_i$ in $D(t)$ do
    if $d_i > T(t)$ then
        Add $v(i)$ to the $L'(t)$
    end if
end for
$tip_1 = random\_select(L'(t), 1)$
$tip_2 = random\_select(L'(t), 1)$
return $tip_1$, $tip_2$
```

7.4 Experiment Design

This section presents two experiments conducted for the proposed TSA: experiments aimed at estimating the critical parameters of the algorithm, and experiments designed to evaluate the algorithm's performance.

7.4.1 Parameter Estimation

Determine α

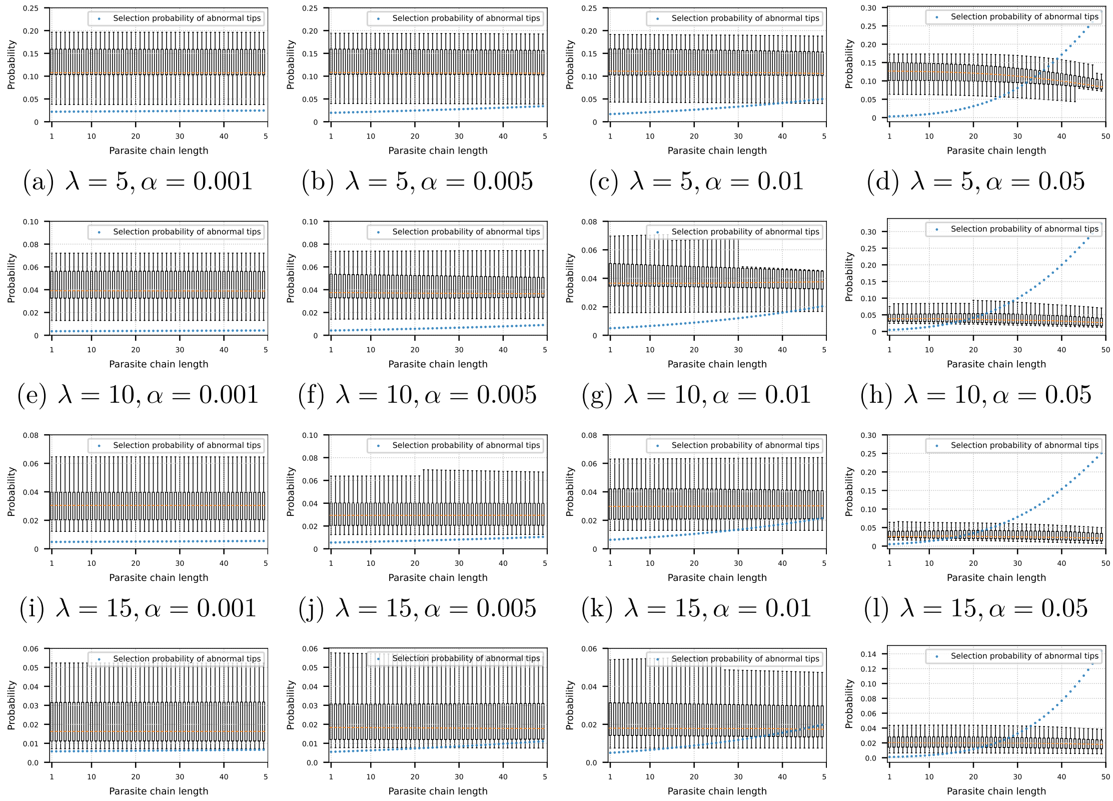
The parameter α of the weighted random walk has an impact on the transition probability between two connected transactions in the tangle. A small value of α results in a very even probability distribution, while a large value of α leads to a scattered probability distribution for the tangle generated by MCMC. However, the effect of α on the probability distribution of the tip in the tangle generated by URTS remains unknown. To determine the most appropriate value of α for S-URTS, we conducted the following empirical experiments.

Table 7.2: Experiment setup: Parameter estimation

Parameters	*Value*
α	0.001, 0.005, 0.01, 0.05
λ	5, 10, 15, 20
N	500
Parasite chain length	from 1 to 200

The experiment was conducted using varying values of λ and α. Some common values, including λ values of 5, 10, 15, 20, and α values of 0.001, 0.005, 0.01, and 0.05, were selected. The tangle consisting of 500 transactions was generated using the URTS algorithm via these λ. Subsequently, parasite chains of varying lengths were attached to a fixed transaction, and the selection probability of tips on the parasite chain and the tips on the normal tangle were calculated. The attachment point was determined based on the maximum distance in the 500-transaction tangle.

The results of the tip selection probability development are shown in Figure 7.4. For a fixed value of λ, as the value of α increases, the selection probability of the tip at the parasite chain becomes more sensitive to the length of the parasite chain. When α is set to 0.001, the increasing rate of the tip selection probability at the parasite chain is slow, and the selection probability of the tip at the parasite chain is always lower than that of the tips at the main tangle. However, when α is set to 0.05, the rate of increase is fast, and the selection probability of the tip at the parasite chain is higher than that of the tip selection probability. Our findings indicate that for each value of λ, the best and most stable performance is achieved when $\alpha = 0.001$. As α increases from 0.001 to 0.05, the tip probability on the parasite chain grows faster. We have also calculated the mean and variance of the difference between the probability of the tip at the parasite chain and at the tangle, and the results are presented in Figure 7.5, which shows that for all values of λ, $\alpha = 0.001$ has a higher mean value and a smaller variance value compared to other values of α. This indicates that with $\alpha = 0.001$, it is easier to detect the tip at the parasite chain.



(a) $\lambda = 5, \alpha = 0.001$ (b) $\lambda = 5, \alpha = 0.005$ (c) $\lambda = 5, \alpha = 0.01$ (d) $\lambda = 5, \alpha = 0.05$

(e) $\lambda = 10, \alpha = 0.001$ (f) $\lambda = 10, \alpha = 0.005$ (g) $\lambda = 10, \alpha = 0.01$ (h) $\lambda = 10, \alpha = 0.05$

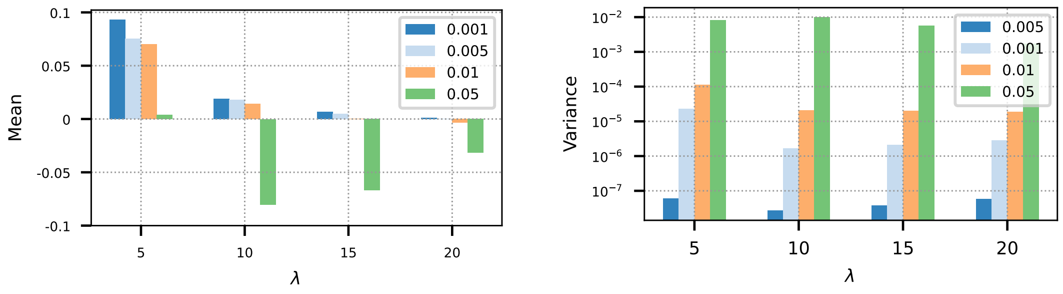
(i) $\lambda = 15, \alpha = 0.001$ (j) $\lambda = 15, \alpha = 0.005$ (k) $\lambda = 15, \alpha = 0.01$ (l) $\lambda = 15, \alpha = 0.05$

(m) $\lambda = 20, \alpha = 0.001$ (n) $\lambda = 20, \alpha = 0.005$ (o) $\lambda = 20, \alpha = 0.01$ (p) $\lambda = 20, \alpha = 0.05$

Figure 7.4: The selection probability of the tip at the main tangle and at the parasite chain

Determine threshold T

The minimum probability in the probability distribution of tips is influenced by the value of λ. Generally, the threshold value T decreases as the number of tips increases. In order to accommodate the arrival of nodes with different λ values, we derive the minimum threshold for tip addition when normal, using the same calculation criteria. If the tip selection probability falls below the threshold, that tip is deemed abnormal. We set $\alpha = 0.001$, generate the tangle using URTS with various λ values: 5, 10, 15, 20, and calculate the minimum selection probability of the tip distribution each round. We then calculate the moving average of the minimum selection probability. Once the moving average value stabilizes and converges, we set it as the threshold for that λ value. Figure 7.6 shows that after 600 messages, the lowest value of the tip is essentially stable around 0.035. Therefore, we adopt the corresponding value of 0.035 as the threshold for abnormal tips for $\lambda = 5$. Using the same method, we calculate that the thresholds for λ values of 10, 15, and 20 are 0.015, 0.01, and 0.007, respectively.



(a) Mean

(b) Variance

Figure 7.5: The mean and variance of the probability difference between normal tip and parasite chain tip

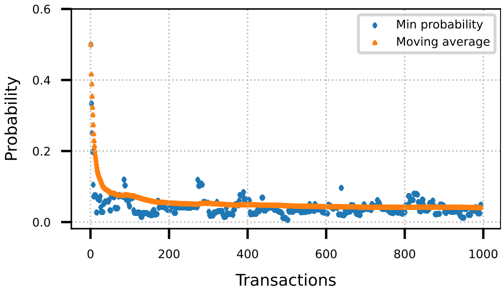


Figure 7.6: The moving average of the minimum tip selection probability

7.4.2 Algorithm Evaluation

The performance evaluation experiments comprise two aspects: scalability and security. In the scalability test, we generate tangles with varying TSAs and parameter settings, and collect data on the tips number and time consumption of tangle generation. Additionally, we analyze the computational complexity of these TSAs. In the security test, we attach parasite chains of varying lengths to the tangle and calculate the selection probability of tips at these parasite chains.

Scalability

We compare the scalability of our proposed algorithm, S-URTS, with two other algorithms, namely, URTS and MCMC, with α values of 0.001 and 0.05. The α value of 0.001 for MCMC was determined through empirical experiments, while the α value of 0.05 was found to be highly sensitive to abnormal structures. Throughout the remainder of this chapter, we will refer to MCMC with $\alpha = 0.001$ as MCMC1 and MCMC with $\alpha = 0.05$ as MCMC5. The experimental setup is presented in Table 7.3.

Table 7.3: Experiment setup: Scalability

Items	*Value*
TSA	URTS, MCMC1, MCMC5, S-URTS
λ	5, 10, 15, 20
N	10^4

Table 7.4: Experiment setup: Security

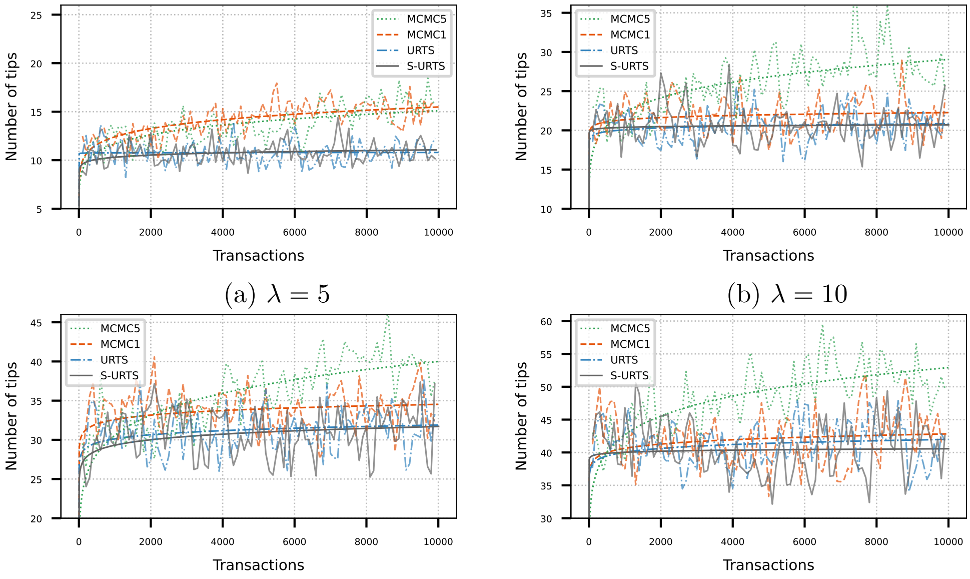
Items	*Value*			
N	500			
α	0.001			
TSA	URTS, MCMC1, S-URTS			
Parasite chain length	from 1 to 200			
λ	5	10	15	20
Attaching point index	400	380	330	300

Security

In order to conduct an analysis of the security of the S-URTS, we have employed a rigorous methodology. Specifically, we have attached parasite chains of varying lengths to a fixed site located at the sub-tangle with a size of N=500. The tip selection probability has been calculated through the use of several algorithms, including S-URTS, MCMC1, and URTS. The selection of the fixed site has been based on the maximum difference between two indexes of the transactions on the tangle. It is important to note that if the attachment position is too close to the normal tips, they cannot be detected, as has been previously noted [PKC+20a]. The detailed experimental settings are presented in Table 7.4.

7.5 Evaluation

In the present section, we undertake a comprehensive analysis of the experimental outcomes and compare the proposed S-URTS with other existing TSAs from two distinct perspectives, namely scalability and security. Regarding to scalability, we delve into the development of the number of tips during the tangle generation process, the time taken for tangle generation, and the computational complexity. In terms of security, we scrutinize the tip selection probability of tips at both the main tangle and the parasite chain.



(a) $\lambda = 5$

(b) $\lambda = 10$

(c) $\lambda = 15$

(d) $\lambda = 20$

Figure 7.7: The comparison of the number of tips development

7.5.1 Scalability

The number of tips

The present study involves the analysis of tip counts during tangle generation using different TSAs, namely: URTS, MCMC1, MCMC5, and S-URTS. The raw data and the fitting line of the data of the number of tips are depicted in Figure 7.7, which provides insights into the development trend of the number of tips with different TSAs and λ values. The tip development of S-URTS is found to be similar to that of URTS, wherein the number of tips initially increases and then stabilizes. Moreover, the number of tips of S-URTS during the stable period is also similar to that of URTS. In the case of MCMC1, when λ is 5, the number of tips shows an increasing trend for a tangle size of 10000. For other λ values, the number of tips of MCMC1 initially increases and then stabilizes at a higher value than that of URTS and S-URTS. As for MCMC5, the number of tips always increases and is greater than the other three TSAs. Theoretically, the minimum number of tips is 2*λ, which is achieved by URTS and S-URTS [Pop16]. These experiments demonstrate that the number of tips of S-URTS can be maintained at a stable and low level.

Consuming time

We collect the consuming time for generating the tangle with 10000 transactions and show the results in Figure 7.8. The results show that when λ is set to 5, URTS outperforms the other three algorithms in terms of consuming time, with S-URTS taking the longest time. However, as λ increases, the consuming time of URTS and MCMC also increases.

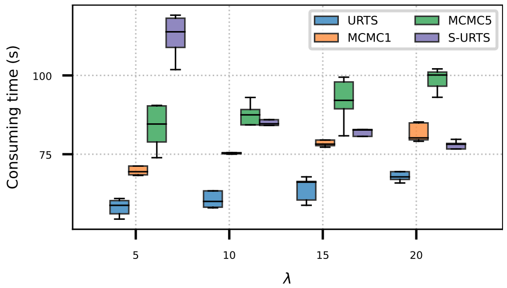


Figure 7.8: Comparisons of consuming time

Specifically, when λ is set to 10, the consuming time of S-URTS is comparable to that of MCMC5, whereas when λ is set to 15, the consuming time of S-URTS is similar to that of MCMC1, but less than that of MCMC5. Finally, when λ is set to 20, the consuming time of S-URTS decreases and becomes less than that of MCMC1 and MCMC5, but higher than that of URTS.

The duration of the batch attaching process has a significant impact on the execution time of S-URTS. Specifically, when the value of λ is relatively small, the number of attaching transactions processed per unit time is correspondingly low. Conversely, as the value of λ increases, the efficiency of S-URTS is enhanced. Despite these fluctuations, the overall execution time of S-URTS remains within an acceptable range.

Computational Complexity

We conducted a comparative analysis of the computational complexity of URTS, MCMC, and S-URTS.

In the case of URTS, the selection of a tip from the tip pool is performed randomly in each step, resulting in a computational complexity of only $O(n)$, n is the number of tips.

For MCMC, the situation is more intricate. MCMC employs a biased random walk and necessitates knowledge of the cumulative weight of each transaction. Based on the definition of cumulative weight, the number of ancestors of each transaction must be calculated, resulting in a computational complexity of $O(|V|^2)$. The subsequent step involves the computation of edge weight. The edge number is denoted as E, and the complexity of calculating edge weight is $O(|E|)$. Similarly, the complexity of calculating transition probability is also $O(|E|)$, as each edge has a transition probability associated with it. The MCMC algorithm for one-time random walk has a complexity of $O(|V|^2 + 2|E|)$. When dealing with a tangle consisting of V transactions, the total calculation time becomes $|V|(|V|^2 + 2|E|)$. This is because each transaction can approve a maximum of two older transactions, and each vertex in the tangle has at most two edges. Therefore, the edge number $|E|$ is equal to or less than $2|V|$. By substituting these values, we can obtain the calculation complexity as $O(|V|^3 + 4|V|^2)$.

The S-URTS algorithm involves two initial steps, namely the calculation of the cumulative weight and transition probability, which are identical to those of the MCMC. The computational complexity of the first step is $O(|V|^2 + 2|E|)$. Additionally, the S-URTS algorithm requires the computation of the selection probability distribution of all tips. The computational complexity of the matrix calculation is $O(|V|^2/\lambda)$. For each round, the computational complexity is $O(|V|^2 + 2|E| + |V|^2/\lambda)$. Assuming an average of λ transactions per round, and a tangle with $|V|$ transactions, it requires approximately $|V|/\lambda$ rounds. The overall computational complexity can be equivalent to $O((\lambda^2 + 1)|V|^3/\lambda^2 + 4|V|^2)$. When λ is large, the computational complexity of the S-URTS algorithm is comparable to that of the MCMC algorithm.

Table 7.5: Computational complexity

TSA	*Complexity*
URTS	$O(n)$
MCMC	$O(\|V\|^3 + 4\|V\|^2)$
S-URTS	$O((\lambda^2 + 1)\|V\|^3/\lambda^2 + 4\|V\|^2)$

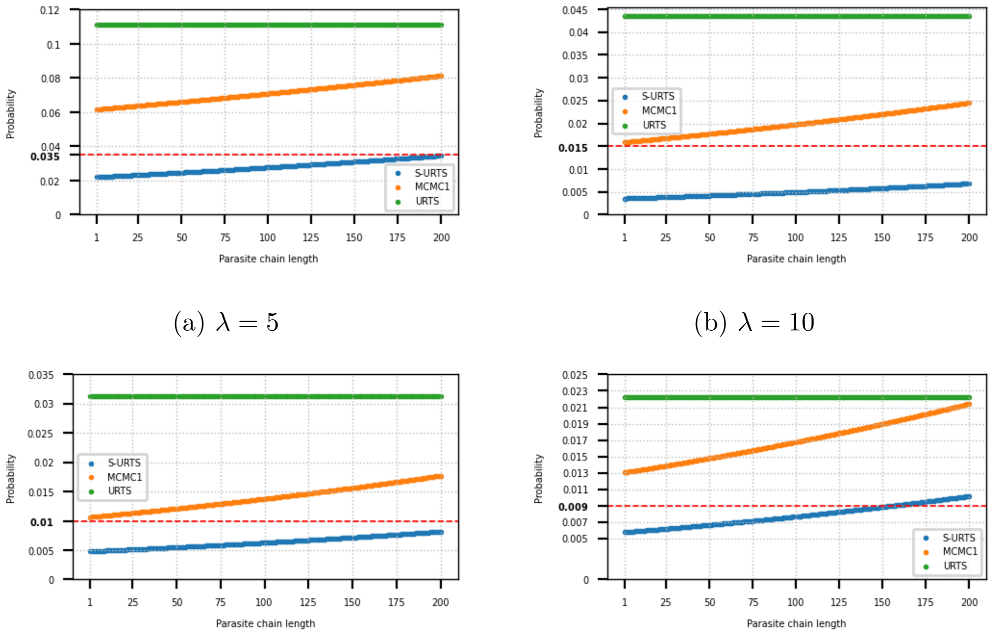
Conclusion

Scalability is a fundamental aspect of blockchain systems, defined as the ability to confirm new transactions promptly as the number of transactions continues to grow. The scalability of the proposed solutions has been evaluated using multiple metrics, including the number of tips (unconfirmed transactions), transaction processing time, and algorithmic complexity.

A critical indicator of scalability is the number of tips in the tangle. If the number of unconfirmed transactions remains low and stable as new transactions arrive, it demonstrates that the system can process incoming transactions in a timely manner. This stability is a hallmark of good scalability. Conversely, an increasing number of tips over time would indicate that the system struggles to confirm transactions promptly, reflecting poor scalability. Our experiments show that the proposed solutions maintain a low and stable tip count even under a high transaction load, proving their ability to handle growing transaction volumes effectively.

Transaction processing time is another key metric for scalability. A shorter processing time indicates the system's ability to handle more transactions per unit of time, showcasing better scalability. Comparative analysis reveals that the proposed solutions achieve faster transaction processing times compared to other methods, enabling the system to process a higher number of transactions efficiently.

The lower computational complexity of the proposed algorithms further supports their scalability. By reducing the computational overhead, the solutions achieve faster transaction processing speeds, ensuring that the system remains efficient even as the transaction volume increases. This contrasts favorably with other algorithms that have higher com-



(a) $\lambda = 5$

(b) $\lambda = 10$

(c) $\lambda = 15$

(d) $\lambda = 20$

Figure 7.9: The comparison of the selection probability of the parasite chain tip for various TSAs (can calculate the MCMC1, MCMC5 and compare them)

plexity and longer processing times, making the proposed solutions more suitable for real-world applications, especially in environments with high transaction loads.

Through our analysis of the number of tips, time required for computation and the computational complexity, we have demonstrated that the S-URTS algorithm is capable of maintaining a stable and low number of tips. Furthermore, we have observed that for larger values of the parameter λ, the consuming time of S-URTS is less than that of MCMC1 and MCMC5. These findings suggest that S-URTS exhibits superior scalability compared to the aforementioned algorithms.

7.5.2 Security

Parasite Chain Attack

In this study, we have affixed parasite chains of varying lengths to the tangle and have subsequently computed the selection probability of the tip on the parasite chain through the utilization of different TSAs. The outcomes of this analysis are presented in Figure 7.9.

The results show that URTS consistently exhibits the highest selection probability across all values of λ. In contrast, the selection probability of S-URTS is significantly lower than that of MCMC1. Furthermore, when λ is set to 5, 10, or 15, the selection probabilities of both URTS and S-URTS fall below the threshold T. Notably, even when the length of the parasite chain is set to 200, the selection probability remains at 0, indicating a secure tangle. However, when λ is set to 20, the tangle becomes vulnerable when the length of the parasite chain exceeds 150. Additionally, as the length of the parasite chain increases, the tip selection probability of MCMC1 increases at a faster rate than that of S-URTS for each λ. Overall, the experimental results suggest that URTS is the most vulnerable TSA, while S-URTS is better than MCMC1 in resisting parasite chain attacks.

7.6 Future Work

This section establishes a theoretical foundation for the S-URTS algorithm, with a primary focus on its scalability and security through simulated testing. However, further work is needed to enhance its practical applicability and to address potential challenges in real-world deployments. Future efforts will concentrate on three main areas: node diversity, network latency, and security threats at the network layer.

7.6.1 Node Diversity

In real-world networks, blockchain nodes often exhibit significant differences in hardware capabilities, processing power, and network bandwidth. This heterogeneity in nodes may impact the overall performance of the algorithm. For resource-constrained nodes, the efficiency of the S-URTS algorithm could decrease, affecting the system's real-time performance and security. Future work will include evaluating the algorithm's adaptability to varying hardware configurations and exploring optimization techniques, such as dynamic parameter adjustments or resource allocation strategies, to enhance the algorithm's robustness in a diverse node environment.

7.6.2 Network Latency

Network latency and communication instability are inevitable in real-world environments, potentially affecting the consensus process of the S-URTS algorithm. Latency can lead to delays in synchronization between nodes, impacting the timeliness of consensus and, under high-latency conditions, may even pose security risks. To address this, the algorithm could incorporate fault-tolerance mechanisms to ensure its resilience under high-latency and packet-loss conditions. Future experiments will test the algorithm's performance under various network conditions (such as high latency and low bandwidth) and identify appropriate network optimization strategies to address these challenges.

7.6.3 Security Threats at the Network Layer

Beyond consensus layer security, blockchain networks face additional threats at the network layer, including transaction censorship and routing attacks. For example, transaction censorship occurs when a lightweight node sends a transaction to a consensus node, which then verifies the transaction's validity before adding it to the blockchain.

In future work, we will explore how optimizing interactions between lightweight and consensus nodes could enhance the system's resilience against these types of attacks and strengthen the network layer's security.

Through these efforts, we aim to build a comprehensive understanding of the S-URTS algorithm's applicability in complex network environments and to support its practical implementation.

7.7 Summary

This section presented a S-URTS algorithm that ensured both scalability and security of a DAG-based blockchain. The proposed algorithm was designed for tip selection, and we further developed algorithms to determine the main parameters α and T for the S-URTS. To demonstrate the scalability and security of the proposed S-URTS, we conducted various experiments. We analyzed scalability in terms of the number of tips, growth trend, time spent on generating tangles, and computational complexity. Additionally, we evaluated security by calculating and comparing the tip selection probability on parasite chains using different TSAs. The experimental results indicated that the proposed S-URTS algorithm effectively stabilizes the number of tips at a very low level, which was lower than the MCMC and essentially equal to the URTS. Furthermore, the time consumption was at a normal level, and the algorithm was capable of resisting parasite chains and avoiding double spending attacks.

Our proposed algorithm would strengthen blockchain-based applications, such as access control and trust management and autonomous systems in IoT. For example, a blockchain-based access control framework for IoT [NZSK] utilizes an encryption algorithm to store access rights on IOTA's Tangle, addressing scalability and transaction cost issues while enabling efficient, fine-grained access control. Our algorithm would further expands this system's capacity to manage access control for a larger number of devices. Additionally, IOTA is used to create a trust overlay for secure information exchange among autonomous vehicles [CMJ], with a tangle architecture integrated with vehicle simulation to assess trustworthiness in decision-making. Our algorithm could enhance the network's ability to support more vehicles.

Overall, the proposed TSA S-URTS algorithm represents a significant contribution to the field of blockchain technology, and its potential applications are numerous.

CHAPTER 8

An Efficient Graph-Based IOTA Tangle Generation Algorithm

IOTA is a recent distributed ledger technology that relies on DAG for its ledger organization. To improve IOTA mechanisms, the state-of-the-art methodology employs graph analysis and, for that, heavily relies on synthetic graph generation. Herein, the most popular generation method simulates IOTA protocol execution. Although this method produces realistic IOTA ledgers, it requires too much memory and time due to repeated random walks on the DAG. In this chapter, we propose an alternative GraGR algorithm designed to generate realistic IOTA ledgers while strongly relaxing memory and timing constraints. The evaluations show that, compared to the state of the art, GraGR can generate a ledger with the same properties with only half of memory and up to 10 times faster.

In the following, we review existing IOTA tangle generation methods in Section 8.2. Section 8.3 introduces IOTA preliminary as a background and presents GraGR; after that, Section 8.4 compares Protocol Simulation based Generator (ProSG)-type generation to tangle generation with GraGR. Finally, Section 8.5 concludes this chapter.

8.1 Introduction

Blockchain is a popular technology that features a decentralized and immutable data ledger [Nak08] with a distributed consensus mechanism. It shows huge potential in areas, where several independently operating authorities work together, such as finance, supply chain management and IoT. However, most current blockchain or DLT exhibit some weaknesses [ZXD^{+}18], such as limited transaction processing speed. In the traditional blockchain it is caused, among others, by the data structure choice: since a chain offers exactly one block that new transactions (or a new block) can be “attached” to, to achieve consistency, either the number of nodes upholding this particular block has to be limited, or complex agreement/consensus between all such nodes at the moment of attachment is required. Both effects limit the transaction throughput [HDM$^+$19].

A solution to improve the scalability of DLT is to use a DAG data structure. Among the DAG-based DLTs, one of the most recent and prominent is IOTA [Pop16]. Here, the incoming transactions can be attached to any leaf node (called “tip”), promising an increased transaction processing speed. However, the speed depends on several factors: an empirical analysis of an operational IOTA data structure (called “tangle”) reveals that the actual processing speed is not as high as expected [GXHD20a], pinpointing the relative complexity of the tangle as the main challenge area. To address this and to improve the performance, graph topology analysis becomes key, as it allows to develop better-suitable, faster algorithms for transaction processing.

For a comprehensive graph analysis, many sample tangles are required. Such samples can be obtained from either real or synthetic data. The problem with real data is limited diversity and availability. The main problem with generated tangles is realism. The common methodology is to follow IOTA protocol for arriving transactions: concretely, after an initiation to a single-vertex DAG, the generator, e.g., IOTA node binary, is subsequently fed with incoming transactions, either taken from a recorded trace or from a stochastic arrival process. Per default, IOTA employs random walk on the transpose graph of its DAG, i.e., from the root to the tips for each incoming transaction, i.e., for each step in the transpose graph, the walker calculates the transition probability for all candidate attachment points. Alas, this transition probability cannot be reused, because the cumulative weight and edge weight change, once the incoming message is attached to the tangle. In a nutshell, this method keeps the whole tangle data structure in memory, including cumulative weight of each message and, potentially, additional graphical information. We refer to these approaches as ProSG. Albeit delivering realistic IOTA tangles, ProSG is not optimized for efficient research data generation and requires a lot of time and memory [XGHD22], in particular under a bursty message arrival. Hence, a more efficient tangle generator is crucial to streamline research and development activities.

This chapter proposes a novel GraGR for realistic IOTA tangle generation. GraGR does not need to calculate the transition probability at each step. Instead, with additionally provided, expected in- and out-degree distributions, GraGR can generate a representative IOTA tangle while limiting memory and timing requirements. Our main contributions are:

- We propose the first IOTA tangle generation algorithm that does not rely on costly random walk approaches.
- We conduct a comprehensive evaluation and analysis of the performance of GraGR in comparison to ProSG type of methods and show that GraGR generation is both correct and efficient.

8.2 Related Work

There are two known ways to obtain IOTA tangles to analyze and improve IOTA system performance. Because they only differ in the used dataset, but rely on the IOTA protocol for message processing, we generally refer to these both methods as ProSG.

The first is to use a synthetic message sequence under IOTA typical message processing. One of such methods is *TangleSimulator*[1]. Starting from a genesis message, *TangleSimulator* generates a Poisson message sequence and adds each message to the tangle using IOTA-typical random walk. A well-known analysis of basic properties of IOTA tangles, such as cumulative weight, number of tips for different TSA versions, etc., also uses this method to create different tangles [KSG18a, KSP$^+$19a]. Similarly, security analysis of IOTA tangles [PKC$^+$20b, GM22b] utilizes tangles generated this way to study parasite chain attacks. Authors in [Vri19] study tangle parameter influence on a so-called large weight attack on an IOTA tangle in a real network. They generate the tangle with a Python library and simulate the large weight attack. An analysis of the TSA properties in [XGHD22] also uses this method and states that it is rather ineffient, in particular for a bursty message arrival.

The second way is to extract ledger data from the operational IOTA ledger, called IOTA mainnet/devnet tangle. An analysis of the real transaction speed in IOTA extracts real tangle data from the IOTA raw dataset, rebuilds the tangle and finds various abnormal structures in the real IOTA tangle that limit the transaction speed [GXHD20a]. However, the raw datasets of real IOTA tangles are of limited diversity and availability.

In contrast to these state of the art approaches, which all rely on the basic IOTA protocol, we aim to generate IOTA tangles based on graph construction mechanisms. Random graph generators are widely used in many fields, such as social networks, biological networks and Internet studies [DT19].

A DAG generation algorithm was proposed in [TK02]. The input is n vertices and m layers. Two vertices are selected from two adjacent layers. A random variable is generated and, if it is smaller than a predefined threshold p, an edge is added from the vertex of the previous layer to the vertex of the latter layer. Common to random graph generators is the fact that the generated graph does not follow IOTA tangle's degree distribution. To improve IOTA mechanisms, it is an important requirement to closely follow topological features of real IOTA tangles [KSG18a].

To find out what would be characteristic properties of tangles, authors in [GXHD20a] analyzed and compared real and theoretical tangles in terms of in-degree distribution, longest and shortest path, diameter ratio and edge weight. The in-degree distribution of simulated tangles based on [Pop16] follows a Poisson distribution. The length of the longest and shortest paths in tangles with 1 million messages are similar and about 10^5. The edge weights of these simulated tangles are distributed in a range from 1 to 100.

[1] https://github.com/minh-nghia/TangleSimulator

Table 8.1: Variable definition

Variable	*Definition*
G_k	The k-th DAG
n	The size of the message set
λ	Message arrival rate
v_i^t	Message i with out-degree t
V, V_k^t	Message set, Message set of G_k with out-degree t
l_i	The number of messages in the layer i
D_k	In-degree distribution of G_k
d_i	In-degree of the message i
T_k	Out-degree distribution
n_k^t	Message count with t out-degree in graph k
E	Index difference distribution
e_{ij}	Index difference between i and j

Tangles should always have a single genesis message and the out-degree of all messages in all kinds of tangles is limited to 2 [Pop16].

In summary, the existing IOTA tangle generators can only generate tangles in an inefficient way, while the existing general random graph generation algorithms do not necessarily comply with the identified constraints of IOTA tangles. Hence, we still need an efficient IOTA tangle generator to generate tangles, which meet the in-degree, out-degree, index difference and other prescribed requirements.

8.3 Algorithm Design

In this section, we introduce the idea of our proposal and present the details of GraGR, the new graph-based tangle generation method.

8.3.1 GraGR Algorithm Design

The basic idea of the IOTA tangle construction method in this chapter is inspired from graph generation methods. In contrast to generic random graph generation, we create an algorithm that creates a DAG with some characteristic topological parameters of a simulated IOTA tangle. Specifically, we want the generated structure to exhibit the same in-degree distribution as an additionally provided input distribution, and to respect typical constraints of the IOTA tangle's out-degree, as described in Section 8.2.

We define V is the message set, D_0 is the prescribed in-degree distribution (input to our algorithm), and E is the prescribed index difference distribution. The index difference is the difference between the indices of two connected messages. The index difference indicates an attachment time interval between two connected messages. For all tangles, the index difference should follow a similar distribution. Otherwise, the tangle structure is abnormal, and the attachment order is chaotic.

The in-degree list d_i is generated by D_0, and index difference list e_{ij} is generated from E.

Table 8.1 summarizes all used variable definitions.

Initially, the generated tangle starts with a genesis message, and the new vertices are attached to this genesis message.

The proposed tangle generation algorithm can be divided into two major parts: part I is the *Generation* part, where we generate a DAG following wanted in-degree and index difference distributions, as shown in Code 6. Part II is the *Refinement* part, where we change output Part I using the out-degree distribution and the index difference distribution, as shown in Code 7.

For *Generation* part, Code 6, inputs are message arrival rate λ, number of nodes N, and the prescribed in-degree distribution D_0. The output of Code 6 is DAG G_1. The major steps are as follows:

1. For the first message v_0 in the tangle, assign a random variable l_1 based on the Poisson distribution determined by arrival rate λ, and add l_1 in-degree to this message.
2. Calculate the index difference for each edge, and the index of messages connected to v_0 is the message v_0 index 0 plus an index difference value.
3. For the following vertices, associate each message v_i with an in-degree value d_i and add d_i messages to this messages, whereas d_i follows the given in-degree distribution D_0.
4. The index of the message added in step 3, which directly connects to the current message, is the current message index plus a random index difference e generated from the index distribution E.

While the output of Algorithm 6, G_1, is a DAG with basic properties of an IOTA tangle, it cannot be considered a realistic IOTA tangle, as it includes too many messages with out-degree 0, and, some of its messages have an out-degree higher than 2. Indeed, in Code 6, we only enforce the prescribed in-degree and add the directly connected children to each message, while out-degree constraints are not respected yet.

The following is the *Refinement* part, Algorithm 7, starts from G_1 and the prescribed out-degree distribution T_0.

1. Calculate the out-degree distribution of G_1 and compare the percentage of each out-degree value of the out-degree distribution T_1 to the prescribed out-degree distribution T_0. Select the messages with unqualified out-degree values, for example, 0 or higher than 2.

2. For all messages with out-degree 0 - except the genesis message - generate two random index difference variables e_{ij}, e_{ih} based on the index difference distribution E. Link v_i to $v_{i-e_{ij}}$, $v_{i-e_{ih}}$.

3. In Step 3, if the index of the added out-degree message is smaller than 0, then link v_i to genesis message v_0.

4. For messages, whose out-degree is bigger than 2, delete the edges randomly, until their out-degree is exactly 2.

5. For messages with out-degree 1, if the prescribed out-degree distribution needs more messages with out-degree 2, then randomly select the messages v_i with out-degree 1, link v_i to a former message $v_{i-e_{ij}}$, the index difference e_{ij}, which is also selected from the index difference distribution E_0.

Algorithm 7 returns DAG G_2, which is a synthetic tangle with prescribed properties.

Algorithm 6 Graph Generation

```
Require: $n$, $V$, $D_0$, $E$, $\lambda$, $T_0$
Ensure: $G_1$
for $v_i$ in $V$ do
    if i == 0 then
        j ~ Poisson($\lambda$)
        $V'=\{v_1,v_2,...,v_j\}$
        add $v_j \in V'$ to $v_0$
    else
        h ~ D
        generate V'=$\{v_{j_1},v_{j_2},...,v_{j_h}\}$
        each $j_h$ = i + e, e ~ E
        add V' to $v_i$
    end if
end for
return $G_1$
```

The GraGR algorithm showcases significant adaptability by supporting various parameter configurations and degree distributions, enabling the generation of diverse tangle topologies. It accommodates multiple parameterized generation models, each designed to produce tangle topologies with specific characteristics. By adjusting degree distribution parameters, the algorithm can create topologies that meet diverse design requirements, making it versatile for a wide range of use cases and environments. This flexibility offers researchers and developers a convenient and efficient method for generating tailored DAG structures, facilitating the evaluation and experimentation of different system configurations.

Algorithm 7 Graph Refinement

```
Require: $G_1$, $n$, $V$, $T_0$
Ensure: $G_2$
for $v_i$ in $V_1$ do
    if Out-degree ($v_i$) == 0 and $i \neq 0$ then
        $e_1$,$e_2$ $\sim$ E
        $v_{j_1}$ = i - $e_1$
        $v_{j_2}$ = i - $e_2$
        if $v_{j_1}$ <0 then
            $v_{j_1} = 0$
        end if
        if $v_{j_2}$ <0 then
            $v_{j_2} = 0$
        end if
        Add $v_i$ to $v_{j_1}$, $v_{j_2}$
    else if Out-degree ($v_i$) >2 then
        Successor ($v_i$) = $\{v_{j_1}$,$v_{j_2}$,...,$v_{j_h}\}$
        random delete edges until Length(Successor($v_i$)) ==2
    else if $n_0^2$ >$n_1^2$ then
        $\Delta n$ = $n_0^2$-$n_1^2$
        random select $\Delta n$ vertices $V'$ from $V_1^1$
        get $V' = \{v_{i_1}$,...,$v_{i_{n'}}\}$
        each $e_{n'} \sim E$, $j_{n'} = i_{n'} - e_{n'}$
        add $v_{i_{n'}} \in V'$ to $v_{j_{n'}}$
    end if
end for
return $G_2$
```

8.4 Evaluation

In this section, we evaluate general ProSG methods and our GraGR proposal in the following aspects: out-degree, in-degree, longest path, shortest path from the genesis message to the latest tip, diameter ratio, index difference and costs, i.e., time and memory consumption. Since the realism of ProSG is not questioned, we use it as a reference for topological parameters. In contrast, we want to have a better performance.

8.4.1 Experiment Setup

We run all experiments on a computer with Intel Core i5-8265U @ 1.6Ghz CPU and 16GB RAM. All algorithms are implemented in Python 3.8. For the experimental setup, we set $\alpha = 0.01$, which is the default value in the real IOTA network. Then, we vary both λ values, and the number of vertices N in Table 8.2. We generate a number of tangles for each set of parameters, until the statistical error of the reported results is lower than

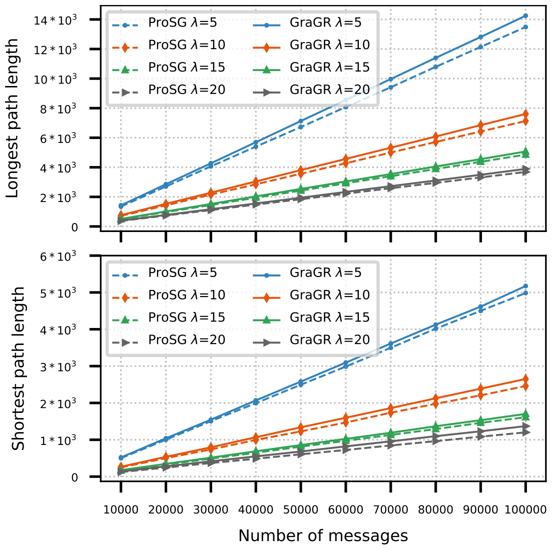


Figure 8.1: Comparisons of path lengths of generated tangles

5%. In practice, the number of generated tangles was around 10 for each parameter set.

Table 8.2: Experiment parameter setup

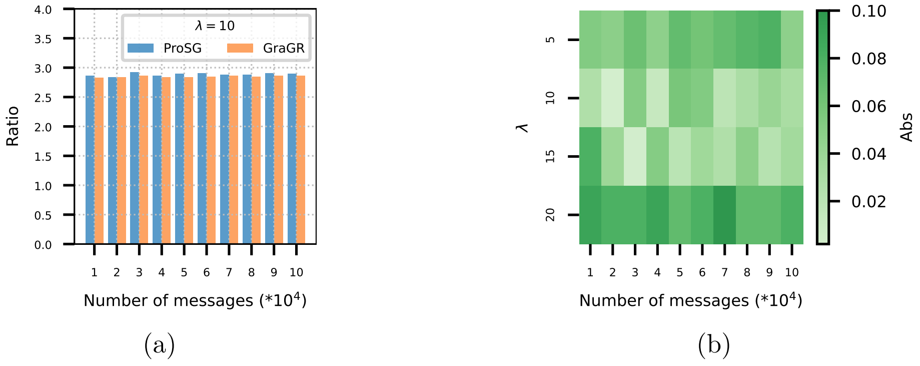
Parameters	*Value*
α	0.01
λ	5, 10, 15, 20
N	$10^4 \sim 10^5$ with a step-size $= 10^4$

8.4.2 Results

Path Length

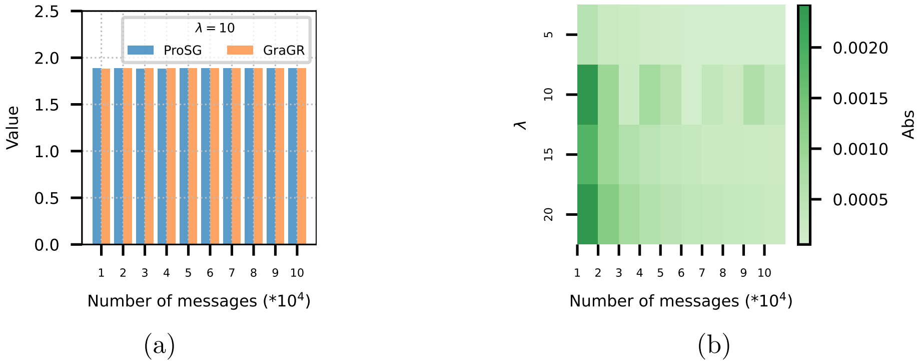
The determined shortest and longest path lengths from the genesis message to the latest tip of the tangles generated by ProSG and GraGR are shown in Figure 8.1.

As can be seen in Figure 8.1, with the number of messages going up, the lengths of the longest and shortest paths increase. For the same number of messages, for smaller λ values the paths are longer. For $\lambda = 1$ (smallest value), the tangle degenerates to a chain. Generally, for bigger λ values, the tangle becomes wider. Note that for the same value of λ, the longest and shortest paths of the tangles, generated by ProSG and GraGR respectively, are similar in length. We conclude that GraGR maintains the path length properties of real IOTA tangles.



(a) (b)

Figure 8.2: The comparison of the tangle diameter ratio



(a) (b)

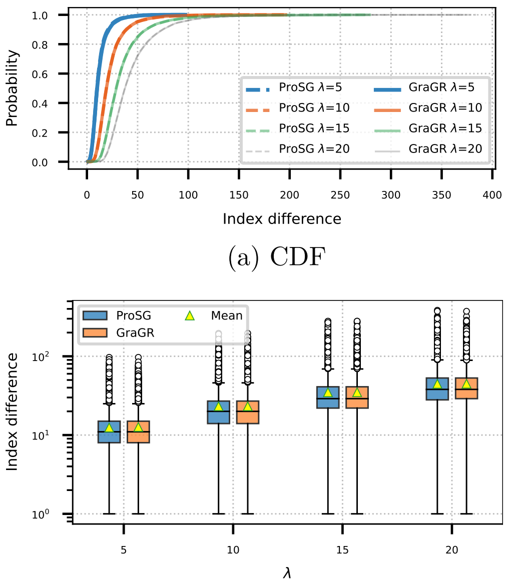
Figure 8.3: The comparison of the tangle in-degree mean

Diameter Ratio

In addition, we calculate the diameter ratio of tangles generated by these two methods and the absolute value of the difference of diameter ratios. We define diameter ratio as the longest path length divided by the shortest path length. Semantically, this term is indicative of the shape of the tangle. A bigger diameter ratio indicates the tangle is wider, while a smaller diameter ratio means that the tangle becomes like a narrow band.

The results are shown in Figure 8.2.

For diameter ratio comparison, we use $\lambda = 10$ as an example, as the results are similar for different λ. In Figure 8.2a, the difference between the diameter ratio of the tangle generated by the two methods is very small. Figure 8.2b shows the diameter ratio difference of the experiments. Most difference values are around 0.06. This comparison indicates that the tangles generated by GraGR and ProSG have quite similar diameter ratios. Hence, we conclude that GraGR generates tangles of realistic shapes.



(a) CDF

(b) Boxplot

Figure 8.4: The comparison of the index difference

Out- and In-Degree Distributions

Just like ProSG methods, which use IOTA TSA, by the strict construction of GraGR in Code 7, the out-degree of each message in the constructed tangle exactly follows the prescribed out-degree distribution, but without employing TSA.

We now evaluate the in-degree distribution of the generated tangles. Specifically, we measure mean and absolute difference values of the in-degree mean of tangles generated by the two methods, as shown in Figure 8.3. Using $\lambda = 10$ as an example, Figure 8.3a shows that the average in-degree of tangles generated by two methods is essentially the same. We show the absolute value of difference of in-degree mean in Figure 8.3b. For higher numbers of vertices (e.g. more than 10000), the difference value is slightly larger. However, the maximum difference is under 0.002, which is still small. As the number of vertices grows, the difference of the in-degree mean becomes smaller, and, when the number of vertices is 100000, the difference is still below 0.0005. Note that the in-degree mean difference remains stable for changing parameter settings. These findings confirm that the tangles generated by ProSG and GraGR have similar properties in terms of in-degree.

Index Difference

We calculate the index difference in the generated tangles and compare their distributions.

Results are presented as CDF in Figure 8.4a. Note the good match of the index difference distribution of the tangles generated by ProSG and GraGR for the same message arrival

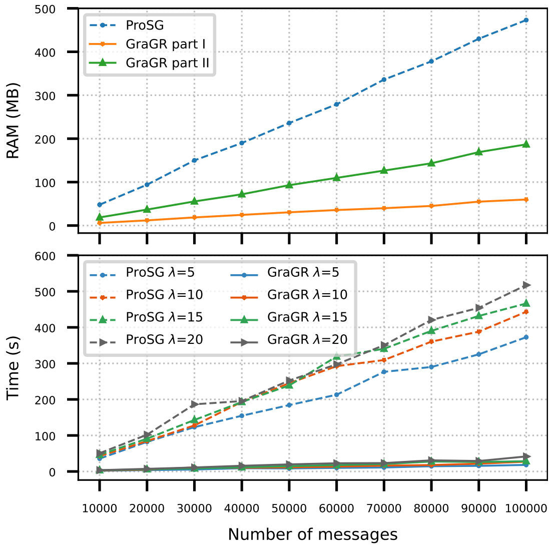


Figure 8.5: The comparison of the consumed time and memory

rate respectively. Depending on λ, the proportion of high index difference values (more than 50) rapidly decreases: while for $\lambda = 5$, there is a very significant proportion of index differences underneath 25, for $\lambda = 10$, the same proportion includes values underneath 50 and for $\lambda = 15$ and $\lambda = 20$ - values under 100.

Figure 8.4b shows this phenomenon more clearly. The index difference values of the tangles generated by ProSG and GraGR are almost the same. The index difference values are distributed in a small range, for example, most index difference values of the tangles with $\lambda = 5$ are mostly distributed in the range (1,20). As λ increases, the median and mean value of the index difference distribution also increase. Overall, however, the index difference in the tangles produced by ProSG and GraGR are similar.

Runtime and Memory Cost

We evaluate both ProSG and GraGR approaches in terms of runtime and memory consumption required when generating tangles of the respectively same size. We run experiments for different tangle sizes N and using different message arrival rates λ. We record the elapsed time and the required memory. The results are shown in Figure 8.5.

First of all, we observe that λ has almost no effect on memory consumption. Hence, we chose $\lambda = 10$ as an example. The upper plot in Figure 8.5 shows the consumed memory. As expected, bigger graphs (more vertices) require more memory; the relationship is essentially linear. To better understand GraGR, we present the consumed memory separately for both of its phases. GraGR Refinement part consumes approx. 3 times more memory than its Generation part, because Refinement needs to calculate the out-degree distribution of the tangle. Comparing GraGR to ProSG, we observe that

GraGR consumes only half of the memory required by ProSG for the same size of the generated IOTA tangle.

The second plot in Figure 8.5 describes the runtime duration of both generators for a tangle of the same size. Again, as expected, with the increasing number of vertices, both ProSG and GraGR require more time; the relationship is, again, essentially linear. However, the runtime of ProSG grows significantly faster than that of our algorithm. Generating a tangle with the same number of vertices, GraGR is up to 10 times faster. Also note that the message arrival rate has a relatively low impact on GraGR execution time.

Overall, GraGR is way more efficient. From the observed trend, and expressing it another way around, on a platform with 1024MB of memory, to produce a tangle with 1 million vertices, the state of the art ProSG requires from 3000 to 5000 seconds, depending on the message arrival rate, while GraGR finishes the task in only 300 to 500 seconds. Given the second-only duration of tangle production for smaller size tangles, with GraGR synthetic tangles can be produced on the fly, without the need to store the tangles for latter analysis. This significantly speeds up the experimentation.

8.5 Conclusion

In this chapter, we propose a novel IOTA tangle generation algorithm. Instead of following IOTA protocol like all existing generators (aka ProSG), our proposal, GraGR, manipulates the topology of a generated random DAG, until its properties fulfill requirements on IOTA tangles. While GraGR delivers results topologically equivalent to ProSG, GraGR consumes only about half of the memory and is up to 10 times faster.

One of the key contributions of this work is the development of an efficient simulator for generating tangle topologies without relying on traditional random walk methods. This innovation addresses significant computational challenges, particularly for scenarios involving extensive parameter studies or experimentation. GraGR directly generates tangle topologies based on pre-defined parameters, enabling researchers to efficiently study the structural properties of the DAG. The efficiency of this simulator is particularly advantageous for research purposes, allowing for rapid evaluation of whether the generated topology meets the desired design objectives. This approach not only advances the methodology for studying DAG topologies but also paves the way for more efficient and scalable implementations of tangle-based systems.

The main difference between ProSG and GraGR is the reliance of ProSG on repetitive reverse random walks. Our evaluations suggest that such random walks are indeed the main contributor to the performance gap of ProSG.

In general, the new tangle generation algorithm provides the research community with an easier way to yield experimental tangles for DAG DLT research. For the first time, GraGR allows on-the-fly generation of realistic IOTA tangles with hundreds of thousands of messages.

We recently found that dPLN distribution may be a better fit [GXHD23b] for tangle degree distribution. While in this chapter, we still rely on the accepted state of the and require Poisson distribution, a generator using dPLN distribution will be addressed in our future work.

CHAPTER 9

A Lightweight Cross-Domain Proximity-Based Authentication Method for IoT Based on IOTA

Nowadays, electronic industry witnesses a massive explosion of offering IoT devices with cellular technology to the market for Machinery Type Communication (MTC). Due to usually unmanned deployments, MTC requires authentication for security reasons before exchanging actual information. Today, IoT cross-domain authentication executed at a blockchain backend side is well studied. However, lightweight proximity-based authentication for cross-domain IoT devices is still lack of consideration. In this chapter, we show the first attempt to solve this problem based on IOTA blockchain technology. Specifically, our solution benefits both from the advantages of IOTA blockchain and the capabilities of MEC so that a lightweight authentication procedure can be achieved by reducing involvements of the heavy backend side. A small in-house prototype system is implemented in order to validate the feasibility of the proposed solution.

The following chapter is organized as follows. Section 9.2 briefly summarizes the literature review. Section 9.3 gives a formal problem statement of our objective. In Section 9.4, we introduce the details of our proposed solution, specifically about how the proximity-based authentication service is built based on IOTA blockchain. After that, our prototype system is introduced in Section 9.5 and Section 9.6 concludes this chapter.

9.1 Introduction

In recent years, the adoption of IoT is rapidly happening in various industry sectors. Today, most people have more than one wearable; increasing number of so-called smart home appliances are observed; most of governments significantly invest on building smart cities and factories (e.g. Industry 4.0 in Germany [Jaz14]). At its heart, IoT devices play an irreplaceable role as the frontier for data collection, transmission and local actuation. With the support from backend clouds, a fully operational loop can be formed without human intervention. With the popularization of IoT devices, MTC will dominate most communication traffic in future networks. Such a phenomenon is further amplified in the forthcoming 5G era because ubiquitous wireless connectivity and high bandwidth with mobility support are assumed as a solved problem.

However, MTC suffers security issues. Firstly, IoT devices are expected to work standalone without regular maintenance (e.g. in open and rural areas), thus easy to be damaged and/or hacked. Secondly, IoT devices are usually resource-constrained, which prevents IoT devices from executing sophisticated mechanisms for self-protection. If IoT devices are compromised without notice, other participants in the system face security risks. Especially when cellular IoT devices come into the whole picture, there will be a significant number of devices deployed at a wild range without regular maintenance. Consequently, self-managed authentication for MTC is a prerequisite before the actual communication starts.

In terms of where the authentication is executed, current proposals can be mainly categorized into two types. The first type is an IoT device authenticated at a backend side. For example, before a smart sensor uploads collected data for further processing, the backend server will have to first authenticate if the IoT device is legitimately identifiable. In this case, the backend side completely determines whether a device is legitimate by verifying some provided authentication information. The backend side can be an intra-/inter-domain authenticator.

The second type is proximity-based authentication, where one IoT device is locally authenticated by another IoT device, instead of forwarding the authentication job to the backend side. Many IoT applications actually can drop into this category. For instance, an electric car (eCar) may want to recharge on-demand at an electric charging station (eCharger). Today some car vendors (e.g. Tesla) deploy their own eChargers, thus authenticating is relatively easy (e.g. directly verifying a shared key secretly stored in both devices) in a single domain. However, it does not have to be the case in the near future because there will certainly be 3rd-party eChargers coming into the market while different car vendors and eCharger providers have to collaborate.

The second type scenario represents a more generalized scenario where two machinery devices from different domains may need to communicate with each other based on a temporal objective. In an idea cyber-physical system, machinery type interactions shall just happen like a social association, where IoT devices assume to be able to “talk” to each other depending on their social relationships, no matter if the two participants are from the same ecosystem. Therefore, it is quite difficult to enable such a temporal interaction between IoT devices themselves if the interaction is cross-domain. It is even harder to enable the two machinery participants to authenticate each other.

This work targets to the proximity-based authentication problem for cross-domain in the latter case where the authentication execution happens locally on a device rather than a backend side. As introduced, the conventional solution is that an authentication request of one device is sent from the visiting domain back to the home domain, wherein the authentication request is processed and then replied back to the visiting domain. After that, the authentication result is returned to the requesting device. First of all, engaging two domains in order to support mutual authentication is a long process involving organizational negotiations. If two domains are not associated (e.g. Tesla's eCharger does not support charging eCars from other companies), cross-domain authentication requests cannot be handled. In addition, this type of solutions not only introduces longer delay, but handling massive concurrent authentication requests could cause a bottleneck issue at the backend side, no matter if blockchain technology is employed [GKS17, HHBS18, WHL18, SLZ$^+$20]. Last but not least, sharing sensitive data to other domains may cause privacy issues.

In this chapter, we propose a lightweight solution for cross-domain proximity-based authentication with reduced involvements from the backend side. Our contributions can be summarized as follows:

- We combine to use IOTA[1] blockchain within MEC to establish a decentralized consortium authentication service network consisting of various IoT service providers;
- We provide detailed IOTA blockchain design including customized transaction definitions and key operation procedures for realizing the proposed proximity-based authentication;
- As an initial step, we implement a small in-house prototype system to validate the feasibility of the proposed solution.

To the best of our knowledge, such an attempt of using IOTA blockchain to build a cross-domain proximity-based authentication is not observed yet.

9.2 Related Work

9.2.1 Single Domain Authentication With Blockchain

For single domain access control, a large number of previous work studied IoT authentication problems where authentication actions happen at a cloud backend side with blockchain. The authors in [GKS17] propose an architecture for scalable IoT access management based on blockchain. A trust region construction scheme is proposed for IoT devices in [HHBS18]. Blockchain for authorization access with smart contract is studied in [AKC$^+$17]. Similar works can be found in [LPDG18, KS18, PTM$^+$18].

[1] https://www.iota.org/

9.2.2 Cross-Domain Authentication With Blockchain

BlockCAM proposed in [WHL18] employs a consortium blockchain to construct a decentralized network with the root Certificate Authority (CA) as the verification nodes. The hash values of the authorized certificates are stored in each block and the verification process only needs to compare whether the hash calculated by the certificate provided by the user is consistent with the hash stored in the blockchain. However, BlockCAM does not remove the root CA, thus it still relies on one trust anchor for certificate authentication. The actual use of blockchain is as of an immutable registry to store the proof of existing certification.

BASA proposed in [SLZ$^+$20] is a blockchain-assisted device authentication system for cross-domain industry IoT. Specifically, a consortium blockchain is introduced to construct trust among different factory domains. Identity-Based Signature (IBS) is exploited during the authentication process. A cross-domain authentication is enabled with the coordination of Private Key Generator (PKG), Area Authentication Service (AAS) and the consortium blockchain. A similar work can be found in [JHS$^+$20], which also uses IBS and cross-domain coordination for signature verification.

However, in both works, heavy executions and interactions are required among the coordination elements while IoT devices pend on until all interaction/authentication procedures are finished at the backend side. Simply put, they belong to the backend side authentication category as introduced before.

9.2.3 Integration of Blockchain and MEC

In [YYS$^+$19], the authors summarized the recent work introducing edge computing to extend the cloud resources and services to be distributed at the edge of the network for blockchain applications. For example, in [GWZ$^+$19], blockchain is integrated with MEC for smart grid network data collection. Another example is in [XZN$^+$18], the authors proposed to outsource the proof-of-work puzzles to MEC for mobile blockchain applications, where an economic approach for edge computing resource management was designed to incentivize collective contributions from mobile edge nodes.

For the authentication problem, in [GHG$^+$19], the authors proposed a distributed and trusted authentication system that combines MEC and blockchain to provide efficient authentication for smart terminals. However, the main goal is to provide an optimized Practical Byzantine Fault Tolerance (PBFT) consensus algorithm. In [AKA$^+$18], the authors proposed to use blockchain combining with fog computing nodes for users who will be authenticated before accessing an IoT device. However, the authentication happens at the backend side but proximity IoT authentication is not solved.

9.2.4 Main Differences

Generally, both intra-/cross-domain authentication solutions built with blockchain technology introduce sophisticated procedures at the backend side, which may cause longer delays and significant overheads. For highly dynamic and on-demand MTC services, this may cause efficiency problems. In addition, most of the existing solutions use classical blockchain technology such as Ethereum [W^{+}14] or Hyperledger Fabric [Cac16] (usually with similar PoW consensus mechanism).

Compared with the literature works, specifically we target to proximity-based authentication and reduce involvements with the backend side. Additionally, the proximity-based authentication shall support cross-domain scenarios in a decentralized, efficient and flexible manner. Last but not least, using IOTA to cross-domain proximity-based authentication problem is not observed according to our literature review. In general, our work can be considered as a complementary proposal in terms of the authentication execution location.

9.3 Problem Statement

In the following discussions, we use 'device' to denote 'CIoT device' for simplicity.

9.3.1 Our Assumptions

First of all, we assume that there are multiple IoT service providers considered, each of which is denoted as SP_i. An IoT service provider represents a service domain, in which every device d_p^i in this domain are under the management of the service provider SP_i. Every device d_a^i is equipped with a cellular interface for accessing mobile network services and a near-field communication interface (e.g. ZigBee or Bluetooth) for proximity interactions.

Secondly, we assume that there is no 3rd-party who can act as a trust anchor for the multiple service providers. As a result, a centralized solution is excluded.

Third, we assume that every device d_a^i is already granted to access mobile network services over its cellular interface, which means that every device has full connections to MECs. Note that being able to access MECs via mobile network services does not mean two devices d_a^i and d_b^j from different service domains can authenticate each other.

Last but not least, as mentioned, we assume that authenticating at backend side may risk latency due to possible cross-domain interactions and signaling overheads. Thus, a proximity-based authentication is preferably considered.

9.3.2 Device Registration Approach

In a service domain, every device d_a^i has to register at its service provider (i.e. at SP_i). There are three main options for registration as follows:

The first way is that a device can create an account (with username and password) at the service provider side. This is the most conventional way as how we usual create an account at a website.

The second way is identity-based method as introduced in the previous section, where a device links its public identity (e.g. an email account) to its private key, which is generated from a PKG. Although this option shows promising benefits such that identity-based encryption and digital signature verification can be easily done, cross-domain interactions (e.g. authentications) suffer large overheads with coordinations between multiple PKGs as already shown in the existing solution [SLZ+20].

Our system setting incorporates the third option, which is PKI-based certificate, where every device gets a certificate from a CA. In our case, the CA of a device d_a^i can be just its service provider SP_i. Formally, we denote a certificate of a device d_a^i that is authorized in service domain SP_i as Cert_a^i. The key reason we use the third option is because proximity-based authentication can directly happen between two devices if the verifying device possesses the public key of the corresponding service provider pubK_i who issues the certificate to the device that is being authenticated. Another reason is that with the rapid development of hardware, the compute resource on a device also upgrades a lot so that computational jobs such as certificate/signature verification will not be a burden anymore.

9.3.3 Cross-Domain Proximity-Based Authentication Problem

Based on the assumptions and given the device registration approach, our problem can be formally described as follows.

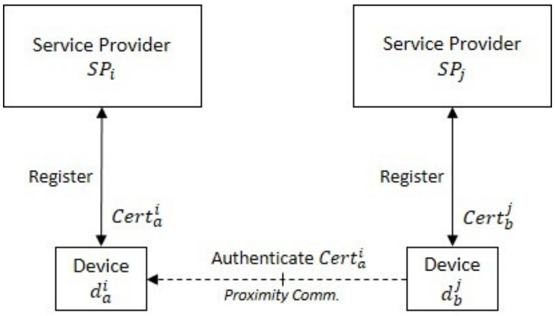


Figure 9.1: Cross-Domain Proximity-Based Authentication Problem

There are two different service providers SP_i and SP_j, each of which manages its own devices and issues certificates to the devices with its asymmetric cryptography key pair $\langle \text{privK}_i, \text{pubK}_i \rangle$. Additionally, device d_a^i and d_b^j register at SP_i and SP_j separately. *Our problem is how device d_b^j can locally fast verify device d_a^i's certificate* Cert_a^i *with less interaction without forwarding the authentication request to the backend side.* An illustration of the problem is also depicted in Figure 9.1.

9.4 Our Solution

9.4.1 System Architecture

Our solution has three main components consisting of a set of service providers ($SP_1, \cdots, SP_k$), devices ($\{d_p^i\}_{i=1}^k$) belonging to corresponding service providers, and a MEC infrastructure that can be operated by multiple mobile network operators.

The service providers can be considered as a device vendor or an OTT (Over-The-Top) companies, and the MECs provided by mobile operators can be considered as underlying resources, on which the application layer builds the cross-domain proximity-based authentication service for the devices. The general architecture is depicted in Figure 9.2.

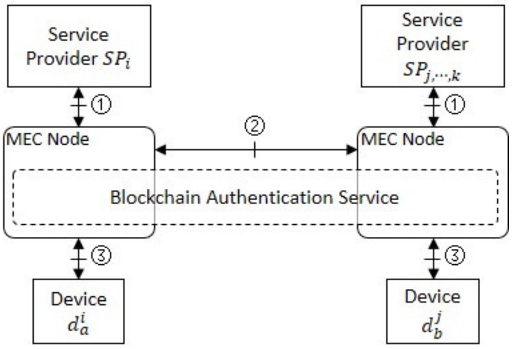


Figure 9.2: General Solution Architecture

Three types of interfaces are introduced in our system. 1) The vertical interface between the OTTs and the MEC resource layer, which is used to deploy blockchain nodes from every service provider to constitute the authentication service; 2) the horizontal interface between MEC nodes, which is used to execute blockchain protocol in order to distribute immutable information. The detailed design of the blockchain will be introduced in the next section; 3) the vertical interface between the devices and the MEC nodes, which is used to access the blockchain authentication service from the end-users.

9.4.2 Main Idea

Every service provider authorizes its own devices by issuing a device certificate. Every service provider publishes at least one piece of information when authorizing a device certificate:

- Cert$_i$: The certificate of the service provider itself, in which the public key information pubK$_i$ that can verify the issued device certificate is included,

onto the blockchain network. After a distributed consensus procedure, the information is immutably synchronized and available on every MEC node.

When a first device (e.g. d_b^j) from one domain (e.g. SP_j) want to authenticate a second device (e.g. d_a^i) from another domain (e.g. SP_i), the first device d_b^j retrieves the published information (i.e. Cert_i) from the local MEC node to validate the provided certificate of the second device (i.e. Cert_a^i). Note that the consideration of certificate revocation that a service provider invalidates a previously issued device certificate will be introduced later.

We can see that the overall procedure does not involve any long authentication execution and interactions happening at backend side and the verifying device executes the authentication locally. In addition, the contact between the two devices are based on direct peer-to-peer communication.

9.4.3 Customized IOTA Blockchain Design

Service providers who want to constitute the cross-domain proximity-based authentication service join in a consortium. The consortium is a permissioned virtual organization, in which every participant is sanctioned to each other after an entering verification process. Note that this is not equivalent to a centralized 3rd-party model. Rather It can happen in a distributed manner as well such as entering with an application process that the majority of existing participants have to approve first. The actual cross-domain authentication service is still built in a decentralized way, where our modified IOTA blockchain network works for.

As we introduced, instead of simply using a traditional blockchain platform, we build our cross-domain authentication service with IOTA blockchain technology. The key feature of IOTA is its lightweight transaction processing manner but without the heavy PoW mining phase. This makes IOTA or its variants considered suitable for IoT applications, wherein tiny, faster, and massive instant transactions are typical cases.

Specific IOTA Transaction Structure

An IOTA blockchain network accommodates transactions and attaches them into a tangle topology, where every single vertex is a transaction object. To the information (i.e., Cert_i) that is published by every service provider, correspondingly, there is a type of transaction, and additionally, there is another type of transaction for publishing revocation log information RevokeLog_a^i against a device certificate. The three types of transactions are listed as follows:

1. SPCert-Tx: Service Provider Certificate Transaction;

2. CertRevoke-Tx: Service Provider Revocation Transaction.

We now introduce their structure definitions respectively whose Python code sample is shown in Listing 9.1.

Listing 9.1: Transaction Definitions (key fields only)

```
class txHeader:
    hash = ''      # Tx hash value of all tx fields, 9
    timestamp = 0  # Tx submission time, 9
    value = 0      # Token value (always 0), 27
    bundle = ''    # Bundle hash pointer, 81
    trunkTx = ''   # The 1st tx existing in tangle, 81
    branchTx = ''  # The 2nd tx existing in tangle, 81
    address = ''   # 0-value address field (random), 81
    attachTag = ''# A user-defined tag, 27

class sPCertTx(txHeader):
    sPCertData = '' # Message buffer, may be fragmented, 2187

class certRevokeTx(txHeader):
    revokeCert = '' # Message buffer, may be fragmented, 2187
```

The two types of transactions share the same transaction header definition. The SPCert-Tx has the sPCertData field, where a service provider encodes its own certificate and publishes with the composed transaction; the CertRevoke-Tx contains a revokeCert field, where a revoked device certificate is encoded in the transaction. All specialized fields here will be serialized in Trytes format and encoded in a message field of a transaction.

If the original message exceeds the maximum length, the transaction will be decomposed into smaller transactions submitted together as a bundle transaction to IOTA network. This is why in the transaction header, there is the bundle field specifying the hash value of a previous transaction in the same bundle.

Since our problem does not involve cryptocurrency tokens, according to IOTA's requirement, our transaction type is always 'zero-value transaction', thus in the transaction definition, the value field is always 0 and the address field specifies no recipient but a random address; furthermore, except specified with value 0, all string fields are in Trytes format and the integer number at the end of the comments tells the maximum field length.

9.4.4 Key Procedures

Given the specialized transaction definitions, the main procedures operating the cross-domain proximity-based authentication are described as follows.

Device Registration

The first procedure is the registration procedure, which is depicted in Figure 9.3.

The first step is that a device sends a registration request consisting of its profile including its pubK$_a^i$ as its ID_a to its IoT service provider SP_i (its privK$_a^i$ is secretly kept locally).

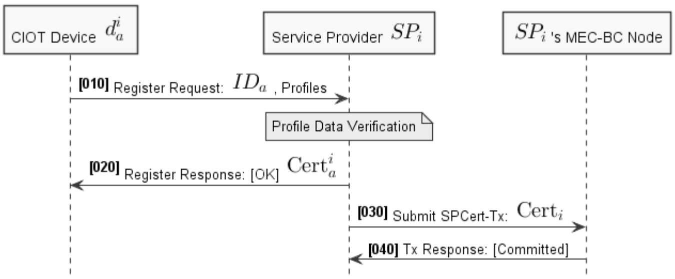

CIOT Device d_a^i
Service Provider SP_i
SP_i 's MEC-BC Node
[010] Register Request: ID_a , Profiles
Profile Data Verification
[020] Register Response: [OK] Cert_a^i
[030] Submit SPCert-Tx: Cert_i
[040] Tx Response: [Committed]

Figure 9.3: Device Registration Procedure

The profile data may contain other fields such as a serial number, manufacture date, model and so on.

Next, the IoT service provider checks the attributes in the profile; if valid, a certificate with a digital signature is issued with specifying an expiration date information as well. Meanwhile, a transaction SPCert-Tx is submitted to an IOTA node deployed at MEC by the service provider. After the transactions are successfully processed by the IOTA network, the commission results are returned. Note that SPCert-Tx does not have to be submitted after every registration if an identical secret material is used repeatedly ot issue certificates; otherwise, every certification has a corresponding secret material behind it.

Service Access and Proximity-Based Authentication

A device d_a^i from one domain sends a service request including its profile and certificate (Cert_a^i) to a second device d_b^j belonging to another domain SP_j. The second device parses the domain information from the first device's certificate (i.e. Cert_a^i).

The second device sends a read request to an IOTA node deployed at MEC to retrieve information of the corresponding service provider SP_i's certificate Cert_i and possibly the latest revocation information related to the certificate of the first device. The IOTA node at MEC responds to the second device with lookup results.

If all information is available, the second device checks if the certificate of the first device was revoked with the found $\mathrm{RevokeLog}_a^i$; if not, the second device then checks if the certificate of the first device is valid by using the service provider's certificate retrieved from the IOTA node; if so, the authentication result is successful; otherwise, failed.

The second device responds the service request with the authentication result and the service access can proceed (if the result is [OK]). The second procedure is depicted in Figure 9.4.

Device Certificate Revocation

A service provider shall be able to manage the certificates issued from it. An important task is to revoke a certificate considered as invalid when necessary, even if the certificate

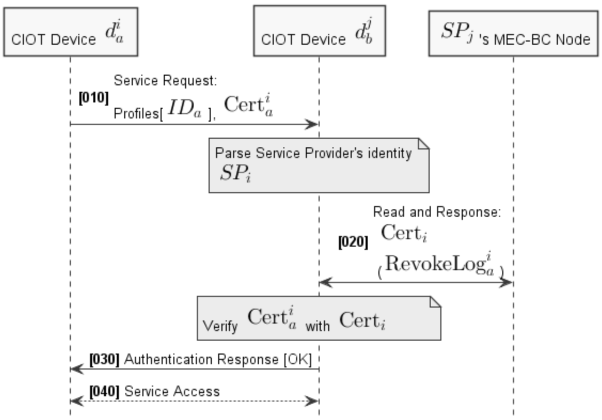

CIOT Device d_a^i
CIOT Device d_b^j
SP_j 's MEC-BC Node
[010] Service Request: Profiles[ID_a], Cert_a^i
Parse Service Provider's identity SP_i
[020] Read and Response: Cert_i ($\mathrm{RevokeLog}_a^i$)
Verify Cert_a^i with Cert_i
[030] Authentication Response [OK]
[040] Service Access

Figure 9.4: Proximity-Based Authentication Procedure

is still before its original expiration date.

A promising feature of IOTA is that a transaction has an AttachTag field as in Listing 9.1, which can be used to specify extra public information regarding the transaction. This helps to quickly search and locate interested transactions without first downloading and decoding a transaction, which is the usual way when using traditional blockchain such as Etheruem.

Specifically, a service provider composes a CertRevoke-Tx, where the AttachTag field specifies the revoked device certificate information and the message field encodes the actual device certificate data, which can be verified. After the commission is done, the service provider gets a confirmation from its IOTA node. The revocation procedure is depicted in Figure 9.5.

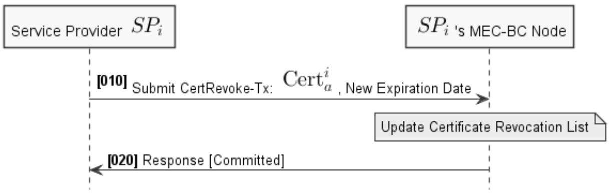

Service Provider SP_i
SP_i 's MEC-BC Node
[010] Submit CertRevoke-Tx: Cert_a^i , New Expiration Date
Update Certificate Revocation List
[020] Response [Committed]

Figure 9.5: Device Certificate Revocation Procedure

Note that although submitting a certificate revocation transaction is simple, some other jobs can be done in parallel so that the efficiency local check at the end user side can be further improved. Specifically, independent to the revocation transaction submissions, every IOTA node in the network can prepare a local Certificate Revocation List (CRL) by searching the AttachTag fields in the ledger records. The local CRL can be a hash table taking device identifiers as the keys so that a device can quickly identify whether or not a certificate is revoked.

9.5 Our Prototype

In order to verify our proposed framework, we have implemented an in-house simplified prototype system. The prototype consists of a small local server farm, two WiFi Access Point (AP)s, and two Raspberry Pi 4 single-board computers:

- **Raspberry Pi 4:** Two Raspberry Pi 4 devices were chosen to simulate CIoT devices. Each device is equipped with Bluetooth for proximity-based communication and WiFi for connectivity with the service providers. The Raspberry Pi 4 provides a quad-core ARM Cortex-A72 processor and 8GB RAM, offering sufficient computational power for cryptographic operations such as certificate validation and signature verification. Its affordability and compatibility with IoT frameworks make it an ideal choice for prototyping.
- **WiFi Access Points (APs):** Two WiFi APs were used to establish a wireless network between the server farm and the Raspberry Pi devices. This ensures seamless communication with the service providers and the IOTA blockchain nodes.
- **Local Server Farm:** A virtualized server environment was deployed to simulate two service providers (SP1 and SP2). The server farm hosts the IOTA blockchain nodes and manages the certification and authentication processes. Each service provider runs on a separate virtual machine, ensuring isolation and realistic management of device registrations and certificate issuance.

9.5.1 System Architecture and Workflow

The architecture is shown in Figure 9.6, wherein the local server farm provides computing resource pool and a virtual network topology is created there. WiFi APs are connected to the servers as an access network for IoT devices. Two Raspberry Pi devices mimic CIoT devices (d_a^1 and d_b^2).

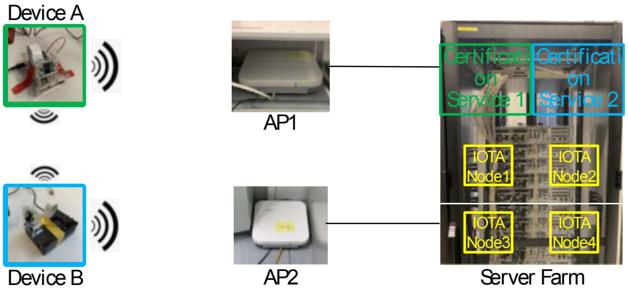


Figure 9.6: A Simplified System Architecture

Firstly, two service providers (SP_1 and SP_2) are deployed as two virtual machines running in the server (in green and blue respectively). The IOTA blockchain network is created by the management tool installed in the server farm. By doing so, we assign two service providers their addresses and accounts including preparing required key materials for certifying devices.

In our prototype, IOTA is configured to launch a 4-node blockchain network representing heterogeneous resources provided from MECs of different operators. The proposed transaction definitions are prepared at the two service providers. Note that, every service provider owns at least one of the four IOTA nodes while the rest of the two can represent other service providers in the same consortium, which are not explicitly shown in this prototype.

Preparation jobs work as follows. Device A (Device B) is configured with the IP address of its own service provider as a default factory configuration. After it boots up and connects to the service provider to do registration (i.e. Device Registration Procedure before). After that, service provider 1 (2) submits an SPCert-Tx to its owned IOTA node. After going through IOTA's distributed consensus protocol (i.e. TSA and site synchronization), the public key certificate is distributed to all the other nodes and stored in the local ledger of every IOTA node.

Proximity authentication works as follows. Device A sends a message containing its certificate to Device B. This is done over a proximity communication channel (Bluetooth interface); after receiving the message, Device B follows the procedure proposed in Figure 9.4 to look up the necessary information from the IOTA node owned by the domain it belongs to. Note that the computational costs of public key certificate verification depend on the digital signature algorithm used at the certification service.

In our experiment, the main interactions of the cross-domain authentication are directly between the two proximity devices, only retrieving the required verification materials is done by reading from an IOTA node at the MEC, which is deployed by the service provider managing Device B. In addition, the deployed IOTA blockchain is only responsible for distributed proof log information and verifying tools (i.e. certificates of service providers), both of which can be done periodically but independently to the authentication events. In other words, they can be prepared before the retrieval requests come.

9.5.2 Scalability and Security Analysis

The current prototype demonstrates scalability up to N devices with the existing server and node setup. However, larger deployments would require optimizations, such as:

- Adding more IOTA nodes to handle increased transaction loads.
- Implementing caching mechanisms for frequently accessed certificates and revocation logs.

The system was tested against several potential attack scenarios:

- **Man-in-the-Middle (MITM) Attacks:** Bluetooth communication was secured using encrypted pairing to mitigate unauthorized interception.

- **Certificate Forgery:** The use of IOTA's tamper-proof blockchain ensures the integrity of issued certificates.
- **Denial of Service (DoS):** The lightweight nature of the IOTA nodes minimizes vulnerability to large-scale DoS attacks, though further enhancements such as rate limiting may be required.

9.5.3 Challenges and Future Work

The implementation of the prototype revealed several challenges that require further attention. One limitation is the computational power of the Raspberry Pi, which, although sufficient for the current implementation, may struggle with highly complex cryptographic algorithms or large-scale deployments. Additionally, environmental factors such as interference can impact the reliability of Bluetooth-based proximity communication. While the IOTA blockchain proved effective for the prototype, further testing is necessary to evaluate its scalability and efficiency in real-world IoT ecosystems. Future work aims to address these challenges by testing alternative hardware, to enhance performance, extending the prototype to support multi-hop proximity authentication, and conducting large-scale simulations to assess scalability and security under malicious attack models.

With the first version of our prototype, we aim to verify the feasibility of the proposed solution. Ongoing work is still being undertaken for performance evaluation on scalability and more testing will be done by including various malicious attack models in our future work.

9.6 Summary

In this chapter, we studied a cross-domain proximity-based authentication problem for IoT devices. In general, our solution provides a local authentication execution between two devices, instead of relying on heavy backend procedures as of existing solutions. Additionally, our solution is built with a lightweight blockchain - IOTA within MEC, which inherits the benefits of the featured technologies. As the first step, feasibility verification was with our in-house prototype and the next step will aim for deeper performance gain evaluations.

CHAPTER 10

Summary and Future Work

In this concluding chapter, we present a comprehensive summary of this thesis. Section 10.1 offers an overview of the topics addressed and the contributions presented. In Section 10.3, we revisit the research questions originally introduced in the thesis's introductory section. Section 10.4 is dedicated to a thoughtful examination of the limitations inherent in our work, concluding with a succinct exploration of potential avenues for future research.

10.1 Summary of Contributions

This thesis primarily delves into the extensive research concerning scalability and security within the realm of the DAG-based blockchain known as IOTA. Our focus has centered on three distinct aspects, each accompanied by comprehensive analysis and the presentation of optimized solutions, as well as an exploration of prospective applications. These facets include: 1) A meticulous examination and characterization of the authentic IOTA tangle; 2) The introduction of a pioneering transaction attachment algorithm designed for incoming transactions; and 3) An in-depth exploration of IOTA's potential applications within MEC.

In our initial phase, we focused on the comprehensive analysis of the properties exhibited by the genuine IOTA tangle. To accomplish this, we undertook the task of acquiring the authentic IOTA database and subsequently reconstructed the IOTA tangle. The findings from this analysis serve as invaluable resources for researchers seeking to gain insight into the authentic properties and performance metrics of IOTA, facilitating their pursuit of further algorithmic optimization. Our investigation extended to an exploration of the in-degree distribution, revealing notable distinctions between the in-degree distributions of the real-world tangle and its simulated counterpart. While the simulated tangle adhered more closely to a Poisson distribution, the real tangle's distribution bore the characteristics of a power law distribution. Moreover, we scrutinized transaction delay in the authentic IOTA and determined that it exceeded initial expectations. Further exploration unveiled the presence of atypical structures within the real IOTA tangle, including instances of blowballing and elongated chains. This analysis also confirmed that the transaction selection algorithm yielded limited influence on the progression of cumulative weight. In an effort to offer a more precise portrayal of the in-degree distribution within the authentic IOTA tangle, we engaged in a fitting process, ultimately concluding that the dPLN provided a superior fit when compared to other long-tail distributions such as LN, Exp, and PL. Our work culminated in the provision of fitting parameters for the dPLN, an achievement facilitated by the development of an innovative fitting algorithm founded on the EM algorithm. Recognizing the dynamic nature of the tangle's generation process and the inherent uncertainty within the tip selection algorithm, we embarked on the development of a theoretical dynamic model aimed at capturing the intricacies of the real IOTA tangle generation process. The insights derived from our research contribute to a more comprehensive and tangible understanding of the evolutionary trajectory of the real IOTA tangle.

Secondly, we introduce an optimized tip selection algorithm tailored to address the challenges posed by high concurrency and diverse transaction scenarios, particularly within the context of IoT deployments. In the course of this thesis, we set forth a dual-tiered approach. Initially, we present a rapid tip selection mechanism designed to manage burst transaction influxes within a DAG-based blockchain framework. Subsequently, we extend this mechanism to propose a secure and scalable tip selection algorithm that maintains a consistent count of unconfirmed transactions. Specifically, our advanced tip selection mechanism eliminates the need for a random walk process to compute tip selection probabilities. Instead, it leverages the transformation of the tangle into an AMC and calculates the probability distribution of stable states. The implementation of AMC not only circumvents the repetition of random walks, thus conserving time and energy resources, but also facilitates batch addition of new transactions. Comparative analysis against conventional random walk-based tip selection algorithms reveals superior performance characteristics. Our proposed algorithm demonstrates diminished transaction delay and reduced processing time, with the advantages becoming more pronounced as the rate of incoming transactions escalates. In scenarios marked by a concurrent influx of transactions, our method stands out as an exemplar of efficiency. Building upon the foundation of the fast tip selection mechanism for burst transaction arrivals, we have made additional refinements and introduced novel features to enhance both the security and scalability of the tangle. This innovative algorithm excels in the selection of abnormal unconfirmed transactions and expedited attachment of new transactions, thus surmounting the prior challenge of reconciling security and scalability. In addition, we have meticulously elucidated the primary parameters employed within this proposed algorithm, delineating both the rationale and computation procedures behind the random walk parameter and the threshold for identifying abnormal transactions.

Thirdly, we introduce a tangle simulator designed to generate the tangle topology without resorting to the random walk process. The simulator operates by initializing predefined parameters, following which it employs graph theory principles to generate the tangle. This approach significantly reduces computational complexity and enhances the speed of tangle generation. Conventionally, the generation of simulated tangles necessitates a minimum of two random walks for each transaction attachment process. In contrast, our innovative graph-based simulator empowers devices with constrained computational resources to expeditiously generate tangles.

Fourthly, this thesis delves into the practical application of IOTA, specifically within the realm of the IoT. IOTA is purposefully designed to cater to IoT needs and has demonstrated its versatility in various use cases. In Chapter xx, we present the development of a lightweight authentication mechanism tailored for devices operating at the mobile edge. This innovative mechanism facilitates the establishment of ephemeral communications between different devices and enables proximity-based cross-domain authentication. Significantly, the entire authentication process occurs at the local devices rather than relying on backend infrastructure. This approach not only streamlines the authentication process by eliminating negotiation requirements and reducing inter-device interactions but also enhances privacy and data security for users. To validate the feasibility of this mechanism, we have implemented an in-house prototype.

10.2 Implications for Various Stakeholders

The findings and contributions of this work extend beyond theoretical advancements, offering practical value to multiple stakeholders in the blockchain and IoT ecosystems. By addressing critical challenges in scalability, security, and system design, this research not only advances the understanding of DAG-based blockchain technologies but also provides actionable insights for platforms, researchers, and end-users. The following sections outline the specific implications for key stakeholders, including the IOTA Foundation, blockchain research institutions, and practical users such as IoT and cloud service providers. These discussions highlight how the proposed solutions can drive innovation, improve system performance, and support real-world applications.

10.2.1 For the IOTA Foundation

By analyzing real-world IOTA tangle data, this work provides the IOTA Foundation with valuable insights into how the tangle evolves and operates in real-world conditions. Unlike theoretical models, this analysis reveals the true nature of the IOTA tangle, identifying areas of potential security risks, deviations from the original design principles, and inefficiencies in the current system. These findings offer a basis for refining the IOTA system, improving its robustness, and addressing vulnerabilities.

Previous research has primarily focused on theoretical aspects or simulations of IOTA; however, this thesis is among the first to delve into real data analysis. The insights generated can serve as a unique resource for the Foundation to understand its system better and guide future development and algorithmic adjustments.

10.2.2 For Blockchain Research Institutions

This work introduces a novel, secure, and highly scalable DAG-based blockchain consensus mechanism, which offers fresh inspiration for researchers. Blockchain institutions can build upon this mechanism, conducting further studies and experiments to enhance it or adapt it for their own use cases.

Leveraging the proposed consensus mechanism, research institutions can design a blockchain system that ensures scalability while maintaining transaction security, which is critical for real-world applications where high throughput and low latency are necessary.

The proposed topology generation algorithm eliminates the need for complex mathematical computations to generate experimental topologies. This feature makes the algorithm a convenient tool for researchers conducting experiments on DAG topologies, saving computational resources and enabling quicker iterations.

10.2.3 Impact on IoT and Cloud Companies

The proposed blockchain-based authentication method provides a cost-effective and efficient solution for scenarios requiring device authentication across different regions. This is particularly beneficial for IoT companies managing devices that operate across regions and need secure cross-domain authentication mechanisms.

By reducing authentication costs and enhancing system security, the proposed approach addresses critical concerns for these companies, paving the way for more secure and scalable IoT deployments.

This work not only advances the theoretical understanding of DAG-based blockchains but also bridges the gap between research and practical applications. The proposed methodologies and insights have the potential to catalyze innovation across various fields, from refining existing blockchain systems to inspiring new use cases in IoT, finance, and other industries.

10.3 Revisiting Research Questions

In Section 1.2, we introduced four key research questions that have served as the foundational pillars of the research presented throughout this thesis. In the concluding chapter of this thesis, we will revisit these research questions and provide a comprehensive summary of the manner in which they have been addressed. Additionally, we will engage in a critical examination of the potential limitations of our research work.

Q1: What does the real IOTA tangle look like, what is the performance of the real IOTA and the dynamic generation model of the real IOTA tangle?

We addressed this question in Chapter 3, Chapter 4, and Chapter 5. In the initial stages of former research, tangle analysis was primarily based on simulated data, lacking insights into the real IOTA tangle. To enhance IOTA's performance, a comprehensive understanding of real tangle properties is indispensable. Our approach involved accessing and parsing the actual IOTA database, which enabled us to convert the tangle data into a readable JSON file. Leveraging the networkx Python package, we reconstructed the tangle and conducted extensive analyses of its characteristics, including in-degree, cumulative weight, and transaction confirmation delay. By sharing our method for real tangle reconstruction in Chapter 3, we've contributed valuable insights. However, it's important to acknowledge certain limitations in this chapter, particularly regarding the precision of tangle in-degree fitting. As the in-degree distribution closely reflects the tangle's generation process, more accurate estimations and fittings of this distribution are required for modeling the dynamic tangle generation process effectively. In Chapter 4, we explored common long-tail distribution fitting and identified that the Double Pareto Lognormal (DPLN) distribution offers a better fit for the in-degree distribution. While this significantly enhanced our understanding of the real tangle's dynamic process, there is still room for improvement in fitting quality. In Chapter 5, we introduced a fitting algorithm based on the EM algorithm to enhance fitting accuracy. Additionally, we utilized SDE to describe the dynamic generation process of the real tangle, further contributing to the field of study.

These contributions collectively advance our understanding of the real IOTA tangle and its dynamic generation process, providing a foundation for further optimization and consensus algorithm improvements. However, ongoing work remains to further enhance the quality of tangle fitting and better characterize its dynamic generation process.

Q2: How to design an efficient and secure transaction selection algorithm running on the resource-constrained device for the DAG-based blockchain?

We addressed this question in Chapter 6 and Chapter 7. In the beginning, the former tip selection algorithms use random walks to select the unconfirmed transaction and attach transactions to the tangle sequentially. Therefore, we propose a TSA without the random walk which can calculate the selection probability distribution of all unconfirmed transactions, then attach the new coming transactions to these transactions with the calculated probability. At first, we propose a tip selection algorithm for burst coming transactions. We set a time window. Then we convert the tangle to the absorbing Markov chain and calculate the absorbing stable state. This absorbing stable state is same to the tip selection distribution of all unconfirmed transactions in this time window. When a bundle of transactions comes to the tangle in this time window, they will be attached to the tangle based on this probability distribution. In this way, the parallel attachment could be achieved the attaching efficiency could be improved and the confirmation delay would be decreased. However, we have found the limitations of this algorithm. When we use a big factor to calculate the probability distribution, the security could be maintained but the number of unconfirmed transactions would be increased. And when a small factor is used, the number of unconfirmed transactions could be controlled, but the security can not be guaranteed. These limitations are solved in Chapter 7. We provide an algorithm that could guarantee security and scalability. After calculating the tip selection probability, the abnormal tips could be selected out based on the predefined threshold value. Then the new transaction could be attached to the tangle randomly. In the end, this algorithm still has some limitations. The calculation and setting of the probability calculation factor and the threshold of the abnormal transaction are based on the empirical method and we still lack a theoretical derivation and analytic solution. The algorithm could only be used in some specific conditions.

These contributions collectively advance our understanding of the real IOTA tangle and its dynamic generation process, providing a foundation for further optimization and consensus algorithm improvements. However, ongoing work remains to further enhance the quality of tangle fitting and better characterize its dynamic generation process.

Q3: How to design an efficient simulator to generate the IOTA tangle fast for the TSA algorithm analysis?

We addressed this question in Chapter 8. While various directed graph generation algorithms exist, they cannot effectively replicate the complex tangle topology structure. Previous tangle simulators also had limitations, as they relied on repeating random walks to generate simulated tangles. To tackle this research question, we introduced a novel simulator that integrates graph generation principles with tangle structure properties, known as the GraGR. GraGR comprises two essential steps. Initially, it establishes directed edges between sites based on predefined in-degree distribution. Subsequently, it refines this arrangement based on the out-degree distribution, aligning it with the predefined out-degree distribution. The output of this second step yields the desired tangle structure. Nonetheless, there are some constraints. The algorithm presented in Chapter 8 generates complete tangle structures but does not simulate their dynamic generation process.

Q4: How to deploy the IOTA in IoT use cases in real life and improve the IoT service performance?

We addressed this question in Chapter 9 through the creation of a lightweight cross-domain proximity-based authentication mechanism. As the IoT continues to expand, an increasing number of devices populate networks, necessitating secure communication through proper authentication. Our designed lightweight cross-domain authentication mechanism enables local authentication before communication, preserving data security and privacy. This solution consists of three main components: a group of service providers, devices affiliated with these service providers, and a MEC infrastructure that accommodates multiple mobile network operators. The authentication process begins with a device sending a registration request to its respective service provider. The service provider verifies the request's attributes and finalizes the registration process. Concurrently, the service provider submits a certificate transaction to an IOTA node. Subsequently, another device from a different domain initiates service access and proximity-based authentication with MEC support. This device forwards service requests and retrieves the certificate from the IOTA node. It then verifies the certificate's validity, completing the authentication process. Additionally, a device certificate revocation mechanism was developed, allowing service providers to issue certification revocation transactions to the IOTA node. The

IOTA node updates the certificate revocation list and confirms the process with the service provider. Nevertheless, several limitations should be noted: 1) The authentication method was tested in a small-scale scenario with a limited number of devices, and its applicability to larger scenarios remains unverified. 2) While feasibility was established, an efficiency and authentication time assessment was not conducted.

10.4 Future Work

In this thesis, we have conducted explorations pertaining to the analysis of DAG-based blockchain IOTA, modeling its generation process, optimizing the consensus methodology, and exploring potential applications. Our efforts have been directed at addressing several critical challenges, particularly those arising from disparities between real tangle and simulated tangle, as well as the existing limitations in achieving a balance between security and scalability through current consensus mechanisms. However, it is important to acknowledge that several challenges and unexplored avenues remain. In this section, we outline these challenges and propose possible directions for future research.

Real-time detection and identification of abnormal structures in tangles

A promising research direction is to analyze the anomalous structures in the DAG-based blockchain, by importing the DAG structure, it can identify what are the anomalous structures and what kind of anomalous structures are they. This technique will improve the security of DAG-based blockchain. Real-time localization and identification of anomalous structures can distinguish which areas have been attacked, thus enhancing the security of the corresponding areas.

In contrast, with a real-time anomaly structure recognition algorithm, we can actually monitor whether the DAG based shows anomalous behavior. When anomalous behavior occurs, action can be taken to avoid further anomalous events and protect the security of the blockchain. Note that the anomalous behavior algorithm does not prevent the node that generates the anomalous behavior from continuing to add transactions to the DAG based blockchain, but it can isolate the anomalous structure by using the anomalous behavior algorithm to prevent other transactions from being added to the anomalous structure.

For example, when a transaction with a large weight of an abnormal structure appears on a DAG based blockchain, the transaction identification algorithm recognizes the anomaly when this transaction structure is added to the blockchain, the current node flags the abnormal transaction structure and adds the newly arrived transaction to the other legitimate transactions so that the abnormal structure cannot be referenced by the other new transactions and thus becomes invalid.

Synchronization of ledger data across diverse devices

A novel research direction that follows our recently proposed transaction selection algorithm is DAG data synchronization across distinct devices based on this algorithm.

This approach focuses on determining the consistency of ledger knowledge between two servers. The goal is to synchronize both the topology data and ledger data of each node while in an asynchronous state. Upon successful data interaction, a consensus is reached among different ledgers.

As outlined in prior sections, our newly devised transaction selection algorithm prioritizes the rapid addition of new transactions to the existing DAG ledger structure, all the while maintaining security and scalability. However, the current implementation exclusively facilitates the swift addition of new transactions on local nodes, without addressing transaction synchronization between different nodes. Failure to synchronize the topology and ledger data among diverse nodes can result in conflicting transactions. To achieve network-wide data synchronization and decentralized data storage, there is an ongoing need for an efficient algorithm to facilitate interactions between nodes.

By adopting an inter-device interaction methodology, we can ensure consistent data synchronization across different nodes, mitigating the risk of conflicting transactions. The device synchronization algorithm will systematically compare the database disparities among various nodes to identify elements that require interactive synchronization. In instances where content or transactions are found to be missing, the node currently lacking said data will request synchronization from neighboring nodes. The ultimate objective is to achieve a harmonized dataset across the network.

Ordering of transactions in a DAG-based blockchain

In a DAG-based blockchain, the simultaneous addition of a multitude of transactions presents a fundamental challenge: how to establish a consistent global order for these transactions. While various transaction ordering algorithms have been employed in different DAG blockchains, a universal ordering algorithm remains absent in a pure DAG-based blockchain, as evident from the literature review. Establishing a global transaction order is crucial as it lays the foundation for providing sequentiality within the blockchain, enabling the deployment and development of smart contracts.

Our previously proposed transaction attachment algorithm, like many other DAG-based approaches, does not inherently consider the order of transaction additions. Unlike traditional chain structures where transactions are uniformly appended one after the other, DAGs involve parallel and unordered transaction additions, potentially leading to varying orders across different nodes. In our approach, transactions are added to the blockchain without a predetermined sequence, prioritizing scalability and security. Consequently, when there is a need to retrieve the transaction order, our system can provide information about transfer records and balance details.

The introduction of a transaction sequencing mechanism offers a solution to this challenge. This mechanism enables the determination of the sequence in which transactions occur within the blockchain. Access to transaction order information empowers the blockchain to support a broader range of services, notably facilitating the execution of smart contracts. For smart contracts to operate effectively, knowledge of the transaction sequence is essential to ensure the accuracy of their execution results.

List of Figures

List of Tables

List of Algorithms

Acronyms

dPLN double Pareto Lognormal. 6, 27, 29, 30, 32–34, 36, 37, 39, 41, 43, 48, 49, 52–57, 59–61, 109, 126

nLP normal-Laplace. 48–50, 52

AAS Area Authentication Service. 114

AMC Absorbing Markov Chain. 63, 65, 67, 68, 70, 126

AP Access Point. 122

BFGS Broyden Fletcher Goldfarb Shanno. 57–60, 133

CA Certificate Authority. 114, 116

CDF Cumulative Distribution Function. 25, 32, 48, 54, 106

COO Coordinator. 18, 26

CPA Cyclic PA. 29, 42

CRL Certificate Revocation List. 121

CW Cumulative Weight. 9, 17, 19, 20, 22, 23

DAG Directed Acyclic Graph. 1–5, 8, 10, 15–17, 27, 39, 40, 42, 63–72, 74, 75, 77, 78, 80, 84, 97–102, 108, 125, 126, 129, 131–133

dApp decentralized Application. 64

DGS Degree Group Size. 30–33, 46–48, 53, 133

DLT Distributed Ledger Technology. 1, 78, 97, 98

EM Expectation-Maximization. 6, 8, 39, 41, 49, 51, 52, 58–60, 126, 129, 133

EW Edge Weight. 9, 22, 23, 25

Exp Exponential. 30, 33, 34, 36, 37, 43, 53, 55–57, 61, 126

GBM Geometric Brownian Motion. 31, 47

GD Gradient Descent. 41, 57

GraGR Graph Generation and Refinement algorithm. 7, 97, 98, 100, 103–108, 130

IBS Identity-Based Signature. 114

IF IOTA Foundation. 1, 16, 19, 23, 27, 29, 32

IoT Internet of Things. 1, 2, 4–6, 8, 15, 24, 28, 64, 66, 77, 78, 97, 111, 112, 114, 115, 118, 122, 126, 127, 130

LN Lognormal. 32, 34, 36, 48, 53–57, 126

MAD Message Attachment Delay. 65, 68, 69, 72, 73, 75

MAE Mean Absolute Error. 53

MCMC Markov Chain Monte Carlo. 2, 5, 10, 12, 18, 78–82, 87, 89–96, 134

MEC Mobile Edge Computing. 7, 111, 113–115, 117, 118, 120, 123–125, 130

MES Master Equation System. 43, 46

MLE Maximization Likelihood Estimation. 33, 49, 53, 57

MS Mainnet Snapshot. 18–24, 133

MSE Mean Square Error. 34

MTC Machinery Type Communication. 111, 112, 115

P2P Peer-to-Peer. 1

PA Preferential Attachment. 28–30, 40–43

PBFT Practical Byzantine Fault Tolerance. 114

PDF Probability Density Function. 32, 33, 48–50

PKG Private Key Generator. 114, 116

PL Power-Law. 29, 30, 33, 34, 36, 37, 42, 43, 53, 55–57, 61, 126

PoW Proof-of-Work. 16, 27, 118

ProSG Protocol Simulation based Generator. 97–99, 103–108

rMSE Root Mean Sqaured Error. 53

rMSLE Root Mean Sqaured Logarithmic Error. 34–36, 53–57, 59, 60, 133

SDE Stochastic Differential Equation. 6, 29, 31, 41, 43, 44, 47, 129

TPS Transactions per Second. 1, 3

TSA Tip Selection Algorithm. 2, 3, 5, 6, 9, 10, 16–20, 22, 23, 26, 63–66, 68, 70–75, 77–82, 86, 89–91, 93–95, 99, 106, 123, 129, 130, 134

TSPD Tip Selection Probability Distribution. 65, 68–75

URTS Uniform Random Tip Selection. 10, 18, 79, 80, 82, 83, 87–96

URW Unbiased Random Walk. 10

Bibliography

[ACA$^+$21] Claudia Antal, Tudor Cioara, Ionut Anghel, Marcel Antal, and Ioan Salomie. Distributed Ledger Technology Review and Decentralized Applications Development Guidelines. *Future Internet*, 13(3):62, March 2021.

[ACL01] William Aiello, Fan Chung, and Linyuan Lu. A Random Graph Model for Power Law Graphs. *Experimental Mathematics*, 10(1):53–66, January 2001.

[AKA$^+$18] Randa Almadhoun, Maha Kadadha, Maya Alhemeiri, Maryam Alshehhi, and Khaled Salah. A user authentication scheme of iot devices using blockchain-enabled fog nodes. In *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE, 2018.

[AKC$^+$17] Michael P Andersen, John Kolb, Kaifei Chen, Gabriel Fierro, David E Culler, and Raluca Ada Popa. Wave: A decentralized authorization system for iot via blockchain smart contracts. *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2017-234*, 2017.

[ASA$^+$23] Serkan Akbulut, Farida Habib Semantha, Sami Azam, Iris Cathrina Abacan Pilares, Mirjam Jonkman, Kheng Cher Yeo, and Bharanidharan Shanmugam. Designing a Private and Secure Personal Health Records Access Management System: A Solution Based on IOTA Distributed Ledger Technology. *Sensors*, 23(11):5174, January 2023.

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[Bai16] Leemon Baird. The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance. *Swirlds Tech Reports SWIRLDS-TR-2016-01, Tech. Rep*, 34:9–11, 2016.

[BCH18] Xavier Boyen, Christopher Carr, and Thomas Haines. Graphchain: A Blockchain-Free Scalable Decentralised Ledger. In *Proceedings of the*

2nd ACM Workshop on Blockchains, Cryptocurrencies, and Contracts, pages 21–33, Incheon Republic of Korea, May 2018. ACM.

[BGP19] Gewu Bu, Önder Gürcan, and Maria Potop-Butucaru. G-IOTA: Fair and confidence aware tangle. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 644–649, April 2019.

[BHP20] Gewu Bu, Wassim Hana, and Maria Potop-Butucaru. E-IOTA: An efficient and fast metamorphism for IOTA. In *2020 2nd Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS)*, pages 9–16, September 2020.

[BL20] Leemon Baird and Atul Luykx. The Hashgraph Protocol: Efficient Asynchronous BFT for High-Throughput Distributed Ledgers. In *2020 International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–7, August 2020.

[BRP18] Michele Bottone, Franco Raimondi, and Giuseppe Primiero. Multi-agent based simulations of block-free distributed ledgers. In *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 585–590. IEEE, 2018.

[BVF18a] Paulo C Bartolomeu, Emanuel Vieira, and Joaquim Ferreira. Iota feasibility and perspectives for enabling vehicular applications. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–7. IEEE, 2018.

[BVF18b] Paulo C. Bartolomeu, Emanuel Vieira, and Joaquim Ferreira. IOTA Feasibility and Perspectives for Enabling Vehicular Applications. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–7, December 2018.

[Cac16] Christian Cachin. Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, volume 310, 2016.

[CFW17] Enrique Calderín-Ojeda, Kevin Fergusson, and Xueyuan Wu. An EM Algorithm for Double-Pareto-Lognormal Generalized Linear Model Applied to Heavy-Tailed Insurance Claims. *Risks*, 5(4):60, December 2017.

[CGWB22] Yinfeng Chen, Yu Guo, Yaofei Wang, and Rongfang Bie. Toward Prevention of Parasite Chain Attack in IOTA Blockchain Networks by Using Evolutionary Game Model. *Mathematics*, 10(7):1108, January 2022.

[Chu16] Anton Churyumov. Byteball: A decentralized system for storage and transfer of value. *URL https://byteball. org/Byteball. pdf*, 2016.

[CMJ] Owen Cutajar, Naghmeh Moradpoor, and Zakwan Jaroucheh. Using IOTA as an Inter-Vehicular Trust Mechanism in Autonomous Vehicles. In *2021 14th International Conference on Security of Information and Networks (SIN)*, volume 1, pages 1–4.

[DDPS21] Sakshi Dhall, Ashutosh Dhar Dwivedi, Saibal K. Pal, and Gautam Srivastava. Blockchain-based Framework for Reducing Fake or Vicious News Spread on Social Media/Messaging Platforms. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 21(1):8:1–8:33, November 2021.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[DT19] Mikhail Drobyshevskiy and Denis Turdakov. Random graph modeling: A survey of the concepts. *ACM Computing Surveys (CSUR)*, 52(6):1–36, 2019.

[DZZ19] Hong-Ning Dai, Zibin Zheng, and Yan Zhang. Blockchain for Internet of Things: A Survey. *IEEE Internet of Things Journal*, 6(5):8076–8094, October 2019.

[EJCF22] Mohammed Elhajj, Hassan Jradi, Maroun Chamoun, and Ahmad Fadlallah. LASII: Lightweight Authentication Scheme using IOTA in IoT Platforms. In *2022 20th Mediterranean Communication and Computer Networking Conference (MedComNet)*, pages 74–83, June 2022.

[FDM$^+$19] Mohamed Amine Ferrag, Makhlouf Derdour, Mithun Mukherjee, Abdelouahid Derhab, Leandros Maglaras, and Helge Janicke. Blockchain Technologies for the Internet of Things: Research Issues and Challenges. *IEEE Internet of Things Journal*, 6(2):2188–2204, April 2019.

[FKCM19a] Caixiang Fan, Hamzeh Khazaei, Yuxiang Chen, and Petr Musilek. Towards A Scalable DAG-based Distributed Ledger for Smart Communities. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pages 177–182. IEEE, 2019.

[FKCM19b] Caixiang Fan, Hamzeh Khazaei, Yuxiang Chen, and Petr Musilek. Towards A Scalable DAG-based Distributed Ledger for Smart Communities. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pages 177–182. IEEE, 2019.

[FKS20] Pietro Ferraro, Christopher King, and Robert Shorten. On the Stability of Unverified Transactions in a DAG-Based Distributed Ledger. *IEEE Transactions on Automatic Control*, 65(9):3772–3783, September 2020.

[FS21] Mohamed Amine Ferrag and Lei Shu. The Performance Evaluation of Blockchain-Based Security and Privacy Systems for the Internet of Things: A Tutorial. *IEEE Internet of Things Journal*, 8(24):17236–17260, December 2021.

[FWKKEBK20] Samuel Fosso Wamba, Jean Robert Kala Kamdjoug, Ransome Epie Bawack, and John G. Keogh. Bitcoin, Blockchain and Fintech: A systematic review and case studies in the supply chain. *Production Planning & Control*, 31(2-3):115–142, February 2020.

[FWS21] Xiang Fu, Huaimin Wang, and Peichang Shi. A survey of Blockchain consensus algorithms: Mechanism, design and applications. *Science China Information Sciences*, 64(2):121101, February 2021.

[Gar09] Diego Garlaschelli. The weighted random graph model. 11(7):073005, July 2009.

[GHG$^+$19] Shaoyong Guo, Xing Hu, Song Guo, Xuesong Qiu, and Feng Qi. Blockchain meets edge computing: A distributed and trusted authentication system. *IEEE Transactions on Industrial Informatics*, 16(3):1972–1983, 2019.

[GKS17] Poonam Ghuli, Urvashi Priyam Kumar, and Rajashree Shettar. A review on blockchain application for decentralized decision of ownership of IoT devices. *Advances in Computational Sciences and Technology*, 10(8):2449–2456, 2017.

[GM22a] Shadan Ghaffaripour and Ali Miri. Parasite Chain Attack Detection in the IOTA Network. In *2022 International Wireless Communications and Mobile Computing (IWCMC)*, pages 985–990, May 2022.

[GM22b] Shadan Ghaffaripour and Ali Miri. Parasite chain attack detection in the iota network. *2022 International Wireless Communications and Mobile Computing (IWCMC)*, pages 985–990, 2022.

[GRW20] Richard Gardner, Philipp Reinecke, and Katinka Wolter. Performance of tip selection schemes in dag blockchains. In *Mathematical Research for Blockchain Economy*, pages 101–116. Springer, 2020.

[GWZ$^+$19] Keke Gai, Yulu Wu, Liehuang Zhu, Lei Xu, and Yan Zhang. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. *IEEE Internet of Things Journal*, 6(5):7992–8004, 2019.

[GXHD20a] Fengyang Guo, Xun Xiao, Artur Hecker, and Schahram Dustdar. Characterizing IOTA Tangle with Empirical Data. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, December 2020.

[GXHD20b] Fengyang Guo, Xun Xiao, Artur Hecker, and Schahram Dustdar. Characterizing IOTA Tangle with Empirical Data. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, December 2020.

[GXHD22a] Guo, Xiao, Artur Hecker, and Schahram Dustdar. A Theoretical Model Characterizing Tangle Evolution in IOTA Blockchain Network. *IEEE Internet of Things Journal*, pages 1–1, 2022.

[GXHD22b] Fengyang Guo, Xun Xiao, Artur Hecker, and Schahram Dustdar. Modeling Ledger Dynamics in IOTA Blockchain. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pages 2650–2655, 2022.

[GXHD23a] Fengyang Guo, Xun Xiao, Artur Hecker, and Schahram Dustdar. An Efficient Graph-Based IOTA Tangle Generation Algorithm. In *ICC 2023 - IEEE International Conference on Communications*, pages 4816–4821, 2023.

[GXHD23b] Fengyang Guo, Xun Xiao, Artur Hecker, and Schahram Dustdar. A Theoretical Model Characterizing Tangle Evolution in IOTA Blockchain Network. *IEEE Internet of Things Journal*, 10(2):1259–1273, January 2023.

[GZS10] Kristian Giesen, Arndt Zimmermann, and Jens Suedekum. The size distribution across all cities – Double Pareto lognormal strikes. *Journal of Urban Economics*, 68(2):129–137, September 2010.

[Hal21] Malka N. Halgamuge. Optimization framework for best approver selection method (BASM) and best tip selection method (BTSM) for IOTA tangle network: Blockchain-enabled next generation industrial IoT. *Computer Networks*, 199:108418, 2021.

[HAÖG22] Vahideh Hayyolalam, Moayad Aloqaily, Öznur Özkasap, and Mohsen Guizani. Edge-Assisted Solutions for IoT-Based Connected Healthcare Systems: A Literature Review. *IEEE Internet of Things Journal*, 9(12):9419–9443, June 2022.

[HCZ$^+$17] David Shui Wing Hui, Yi-Chao Chen, Gong Zhang, Weijie Wu, Guanrong Chen, John C. S. Lui, and Yingtao Li. A Unified Framework for Complex Networks with Degree Trichotomy Based on Markov Chains. *Scientific Reports*, 7(1):3723, June 2017.

[HDM+19] Laurie Hughes, Yogesh K. Dwivedi, Santosh K. Misra, Nripendra P. Rana, Vishnupriya Raghavan, and Viswanadh Akella. Blockchain research, practice and policy: Applications, benefits, limitations, emerging research themes and research agenda. *International Journal of Information Management*, 49:114–129, December 2019.

[HHBS18] Mohamed Tahar Hammi, Badis Hammi, Patrick Bellot, and Ahmed Serhrouchni. Bubbles of Trust: A decentralized blockchain-based authentication system for IoT. *Computers & Security*, 78:126–142, September 2018. cites: hammiBubblesTrustDecentralized2018.

[Jaz14] Nasser Jazdi. Cyber physical systems in the context of Industry 4.0. In *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, pages 1–4. IEEE, 2014.

[JHS+20] Xudong Jia, Ning Hu, Shen Su, Shi Yin, Yan Zhao, Xinda Cheng, and Chi Zhang. Irba: An identity-based cross-domain authentication scheme for the internet of things. *Electronics*, 9(4):634, 2020.

[Kar61] Jack Karush. On the Chapman-Kolmogorov Equation. *The Annals of Mathematical Statistics*, 32(4):1333–1337, 1961.

[Kem81] John G. Kemeny. Generalization of a fundamental matrix. *Linear Algebra and its Applications*, 38:193–206, 1981.

[KG18] Bartosz Kusmierz and Alon Gal. Probability of being left behind and probability of becoming permanent tip in the tangle. Technical report, Technical Report. Accessed: 2018-04-16, 2018.

[KKKR22] R. Lakshmana Kumar, Firoz Khan, Seifedine Kadry, and Seungmin Rho. A Survey on blockchain for industrial Internet of Things. *Alexandria Engineering Journal*, 61(8):6001–6022, August 2022.

[KP13] Dharshana Kasthurirathna and Mahendra Piraveenan. Cyclic Preferential Attachment in Complex Networks. *Procedia Computer Science*, 18:2086–2094, January 2013.

[KS98] Ioannis Karatzas and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*, volume 113 of *Graduate Texts in Mathematics*. Springer, New York, NY, 1998.

[KS18] Minhaj Ahmad Khan and Khaled Salah. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82:395–411, 2018.

[KSG18a] Bartosz Kusmierz, Philip Staupe, and Alon Gal. Extracting tangle properties in continuous time via large-scale simulations. Technical report, Technical Report. Accessed: 2018-08-23, 2018.

[KSG18b] Bartosz Kusmierz, Philip Staupe, and Alon Gal. Extracting Tangle Properties in Continuous Time via Large-Scale Simulations. page 21, 2018.

[KSP$^+$19a] Bartosz Kusmierz, William Sanders, Andreas Penzkofer, Angelo Capossele, and Alon Gal. Properties of the Tangle for Uniform Random and Random Walk Tip Selection. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 228–236, July 2019.

[KSP$^+$19b] Bartosz Kusmierz, William Sanders, Andreas Penzkofer, Angelo T. Capossele, and Alon Gal. Properties of the tangle for uniform random and random walk tip selection. *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 228–236, 2019.

[Kus17] B. Kusmierz. The first glance at the simulation of the Tangle: Discrete model. *IOTA Found. WhitePaper*, pages 1–10, 2017.

[LFY$^+$19] Jiaqi Liu, Luoyi Fu, Yuhang Yao, Xinzhe Fu, Xinbing Wang, and Guihai Chen. Modeling, Analysis and Validation of Evolving Networks With Hybrid Interactions. *IEEE/ACM Transactions on Networking*, 27(1):126–142, February 2019.

[LP75] J. T. Lewis and J. V. Pulè. Dynamical theories of Brownian motion. In Huzihiro Araki, editor, *International Symposium on Mathematical Problems in Theoretical Physics*, Lecture Notes in Physics, pages 294–296, Berlin, Heidelberg, 1975. Springer.

[LPDG18] Dongxing Li, Wei Peng, Wenping Deng, and Fangyu Gai. A blockchain-based authentication and security mechanism for iot. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–6. IEEE, 2018.

[Mit03a] Michael Mitzenmacher. Dynamic Models for File Sizes and Double Pareto Distributions. *Internet Mathematics*, 1(3):305–333, January 2003.

[Mit03b] Michael Mitzenmacher. Dynamic Models for File Sizes and Double Pareto Distributions. *Internet Mathematics*, 1(3):305–333, January 2003.

[MWB$^+$22] Tamara Islam Meghla, Md Whaiduzzaman, Alistair Barros, Md. Julkar Nayeen Mahi, Mehdi Sookhak, Colin Fidge, and Rajkumar Buyya. IOTA-Based Efficient and Reliable Scheme for Internet of Vehicles. In Sazzad Hossain, Md. Shahadat Hossain, M. Shamim Kaiser, Satya Prasad Majumder, and Kanad Ray, editors, *Proceedings of International Conference on Fourth Industrial Revolution and Beyond 2021*, Lecture Notes in Networks and Systems, pages 385–400, Singapore, 2022. Springer Nature.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[Nak09] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. *Cryptography Mailing list at https://metzdowd.com*, March 2009.

[NCK$^+$21] Quan Nguyen, Andre Cronje, Michael Kong, Egor Lysenko, and Alex Guzev. Lachesis: Scalable Asynchronous BFT on DAG Streams, August 2021.

[Nel67] Edward Nelson. *Dynamical Theories of Brownian Motion.* Princeton University Press, 1967.

[NM65] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, January 1965.

[NZSK] Ruka Nakanishi, Yuanyu Zhang, Masahiro Sasabe, and Shoji Kasahara. IOTA-Based Access Control Framework for the Internet of Things. In *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pages 87–95.

[PA17] Pradumn Kumar Pandey and Bibhas Adhikari. A Parametric Model Approach for Structural Reconstruction of Scale-Free Networks. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2072–2085, October 2017.

[PAD20] Pericle Perazzo, Antonio Arena, and Gianluca Dini. An Analysis of Routing Attacks Against IOTA Cryptocurrency. In *2020 IEEE International Conference on Blockchain (Blockchain)*, pages 517–524, November 2020.

[PCAM19] Joon Park, Ruzanna Chitchyan, Anastasia Angelopoulou, and Jordan Murkin. A block-free distributed ledger for p2p energy trading: Case with iota? In *International Conference on Advanced Information Systems Engineering*, pages 111–125. Springer, 2019.

[Pil16] Marc Pilkington. Blockchain technology: principles and applications. In *Research handbook on digital transformations.* Edward Elgar Publishing, 2016.

[PKC$^+$20a] Andreas Penzkofer, Bartosz Kusmierz, Angelo Capossele, William Sanders, and Olivia Saa. Parasite Chain Detection in the IOTA Protocol, April 2020.

[PKC$^+$20b] Andreas Penzkofer, Bartosz Kusmierz, Angelo T. Capossele, William Sanders, and Olivia Saa. Parasite chain detection in the iota protocol. In *Tokenomics*, 2020.

[PLH$^+$21] Kai Peng, Meijun Li, Haojun Huang, Chen Wang, Shaohua Wan, and Kim-Kwang Raymond Choo. Security Challenges and Opportunities for Smart Contracts in Internet of Things: A Survey. *IEEE Internet of Things Journal*, 8(15):12004–12020, August 2021.

[PMC$^+$20] Serguei Popov, Hans Moog, Darcy Camargo, Angelo Capossele, Vassil Dimitrov, Alon Gal, Andrew Greve, Bartosz Kusmierz, Sebastian Mueller, and Andreas Penzkofer. The coordicide. *Accessed Jan*, pages 1–30, 2020.

[Pop16] Serguei Popov. The tangle. *White paper*, 2016.

[PSF19a] Serguei Popov, Olivia Saa, and Paulo Finardi. Equilibria in the tangle. *Computers & Industrial Engineering*, 136:160–172, 2019.

[PSF19b] Serguei Popov, Olivia Saa, and Paulo Finardi. Equilibria in the tangle. *Computers & Industrial Engineering*, 136:160–172, October 2019.

[PSV$^+$22] Nihar Ranjan Pradhan, Akhilendra Pratap Singh, Sahil Verma, Kavita, Marcin Wozniak, Jana Shafi, and Muhammad Fazal Ijaz. A blockchain based lightweight peer-to-peer energy trading framework for secured high throughput micro-transactions. *Scientific Reports*, 12(1):14523, August 2022.

[PTM$^+$18] Alfonso Panarello, Nachiket Tapas, Giovanni Merlino, Francesco Longo, and Antonio Puliafito. Blockchain and IoT integration: A systematic survey. *Sensors*, 18(8):2575, 2018.

[QYB$^+$19] Maoying Qiao, Jun Yu, Wei Bian, Qiang Li, and Dacheng Tao. Adapting Stochastic Block Models to Power-Law Degree Distributions. *IEEE Transactions on Cybernetics*, 49(2):626–637, February 2019.

[Ree06] William J. Reed. The Normal-Laplace Distribution and Its Relatives. In N. Balakrishnan, José María Sarabia, and Enrique Castillo, editors, *Advances in Distribution Theory, Order Statistics, and Inference*, Statistics for Industry and Technology, pages 61–74. Birkhäuser, Boston, MA, 2006.

[RID$^+$23] Syafiqur Rochman, Jazi Eko Istiyanto, Andi Dharmawan, Vian Handika, and Satriawan Rasyid Purnama. Optimization of tips selection on the IOTA tangle for securing blockchain-based IoT transactions. *Procedia Computer Science*, 216:230–236, January 2023.

[RJ04a] William J. Reed and Murray Jorgensen. The Double Pareto-Lognormal Distribution—A New Parametric Model for Size Distributions. *Communications in Statistics - Theory and Methods*, 33(8):1733–1753, December 2004.

[RJ04b] William J. Reed and Murray Jorgensen. The Double Pareto-Lognormal Distribution—A New Parametric Model for Size Distributions. *Communications in Statistics - Theory and Methods*, 33(8):1733–1753, December 2004.

[Ros95] Sheldon M. Ross. *Stochastic Processes.* John Wiley & Sons, February 1995.

[RYS$^+$20] Team Rocket, Maofan Yin, Kevin Sekniqi, Robbert van Renesse, and Emin Gün Sirer. Scalable and Probabilistic Leaderless BFT Consensus through Metastability, August 2020.

[siz] The size distribution across all cities – Double Pareto lognormal strikes - ScienceDirect. https://www.sciencedirect.com/science/article/pii/S0094119010000185.

[SLZ16] Yonatan Sompolinsky, Yoad Lewenberg, and Aviv Zohar. Spectre: A fast and scalable cryptocurrency protocol. *Cryptology ePrint Archive*, 2016.

[SLZ$^+$20] Meng Shen, Huisen Liu, Liehuang Zhu, Ke Xu, Hongbo Yu, Xiaojiang Du, and Mohsen Guizani. Blockchain-assisted secure device authentication for cross-domain industrial iot. *IEEE Journal on Selected Areas in Communications*, 38(5):942–954, 2020.

[SMS$^+$08a] Mukund Seshadri, Sridhar Machiraju, Ashwin Sridharan, Jean Bolot, Christos Faloutsos, and Jure Leskove. Mobile call graphs: Beyond power-law and lognormal distributions. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 08*, page 596, Las Vegas, Nevada, USA, 2008. ACM Press.

[SMS$^+$08b] Mukund Seshadri, Sridhar Machiraju, Ashwin Sridharan, Jean Bolot, Christos Faloutsos, and Jure Leskove. Mobile call graphs: Beyond power-law and lognormal distributions. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 596–604, New York, NY, USA, August 2008. Association for Computing Machinery.

[Sub17] Hemang Subramanian. Decentralized Blockchain-based electronic marketplaces. *Communications of the ACM*, 61:78–84, December 2017.

[TBW19] Etienne Gael Tajeuna, Mohamed Bouguessa, and Shengrui Wang. Modeling and Predicting Community Structure Changes in Time-Evolving Social Networks. *IEEE Transactions on Knowledge and Data Engineering*, 31(6):1166–1180, June 2019.

[TCS$^+$22] Ali Tekeoglu, Chen-Fu Chiang, Saumendra Sengupta, Norman Noor Ahmed, Michael Stein, and Dilip Kusukuntla. Optimized Transaction Processing in Lightweight Distributed Ledger Networks for Internet of Things. In Shiping Chen, Rudrapatna K. Shyamasundar, and Liang-Jie Zhang, editors, *Blockchain – ICBC 2022*, Lecture Notes in Computer Science, pages 117–128, Cham, 2022. Springer Nature Switzerland.

[TK02] Takao Tobita and Hironori Kasahara. A standard task graph set for fair evaluation of multiprocessor scheduling algorithms. *Journal of Scheduling*, 5(5):379–394, 2002.

[Tol79] Richard Chace Tolman. *The Principles of Statistical Mechanics*. Courier Corporation, January 1979.

[VK76] N. G Van Kampen. Stochastic differential equations. *Physics Reports*, 24(3):171–228, March 1976.

[Vri19] L. J. W. Vries. *IOTA Vulnerability: Large Weight Attack Performed in a Network*. B.S. thesis, University of Twente, 2019.

[W$^+$14] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

[WCL23] Zhiwei Wang, Qingqing Chen, and Lei Liu. Permissioned Blockchain-Based Secure and Privacy-Preserving Data Sharing Protocol. *IEEE Internet of Things Journal*, 10(12):10698–10707, June 2023.

[WDW20] Yulei Wu, Hong-Ning Dai, and Hao Wang. Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0. *IEEE Internet of Things Journal*, 8(4):2300–2317, 2020.

[WGYZ09] Li-Na Wang, Jin-Li Guo, Han-Xin Yang, and Tao Zhou. Local preferential attachment model for hierarchical networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1713–1720, April 2009.

[WHL18] Wentong Wang, Ning Hu, and Xin Liu. Blockcam: A blockchain-based cross-domain authentication model. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pages 896–901. IEEE, 2018.

[Woo14] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[WR15] Uri Wilensky and William Rand. *An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo.* Mit Press, 2015.

[WYW21] Jingzhong Wang, Jiahao Yang, and Baocheng Wang. Dynamic balance tip selection algorithm for IOTA. In *2021 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 5, pages 360–365. IEEE, 2021.

[WZ19] Huifang Wang and Zhihong Zhang. A tsgp-based tip search optimization algorithm. In *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 424–427. IEEE, 2019.

[XGH20] Xun Xiao, Fengyang Guo, and Artur Hecker. A lightweight cross-domain proximity-based authentication method for IoT based on IOTA. In *2020 IEEE Globecom Workshops (GC Wkshps*, pages 1–6. IEEE, 2020.

[XGHD22] Xun Xiao, Fengyang Guo, Artur Hecker, and Schahram Dustdar. Fast Tip Selection for Burst Message Arrivals on A DAG-based Blockchain Processing Node at Edge. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pages 1373–1378, December 2022.

[XZN$^+$18] Zehui Xiong, Yang Zhang, Dusit Niyato, Ping Wang, and Zhu Han. When mobile blockchain meets edge computing. *IEEE Communications Magazine*, 56(8):33–39, 2018.

[YW21] Qing Yang and Hao Wang. Privacy-Preserving Transactive Energy Management for IoT-Aided Smart Homes via Blockchain. *IEEE Internet of Things Journal*, 8(14):11463–11475, July 2021.

[YYS$^+$19] Ruizhe Yang, F Richard Yu, Pengbo Si, Zhaoxin Yang, and Yanhua Zhang. Integrated blockchain and edge computing systems: A survey, some research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(2):1508–1532, 2019.

[ZBLN97a] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, December 1997.

[ZBLN97b] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, December 1997.

[ZGK$^+$22] Samman Zahra, Wei Gong, Hasan Ali Khattak, Munam Ali Shah, and Houbing Song. Cross-Domain Security and Interoperability in Internet of Things. *IEEE Internet of Things Journal*, 9(14):11993–12000, July 2022.

[ZMN17] Xiao Zhang, Cristopher Moore, and Mark E. J. Newman. Random graph models for dynamic networks. *The European Physical Journal B*, 90(10):200, October 2017.

[ZXD$^+$18] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International journal of web and grid services*, 14(4):352–375, 2018.

[ZXE$^+$23] Cheng Zhang, Yang Xu, Haroon Elahi, Deyu Zhang, Yunlin Tan, Junxian Chen, and Yaoxue Zhang. A Blockchain-Based Model Migration Approach for Secure and Sustainable Federated Learning in IoT Systems. *IEEE Internet of Things Journal*, 10(8):6574–6585, April 2023.

[ZZSS22] Hongwei Zhang, Marzia Zaman, Brian Stacey, and Srinivas Sampalli. A Novel Distributed Ledger Technology Structure for Wireless Sensor Networks Based on IOTA Tangle. *Electronics*, 11(15):2403, January 2022.