A Scalable and Secure Transaction Attachment Algorithm for DAG-Based Blockchain

Fengyang Guo, *Student Member, IEEE*, Artur Hecker, and Schahram Dustdar, *Fellow, IEEE*

***Abstract*—Blockchain, as an innovative distributed ledger technology, has attracted considerable attention in recent years from both academic circles and industry sectors. Its applications span a diverse range of domains, including finance and the Internet of Things (IoT). However, the scalability of blockchain technology is still a critical limitation with the increasing volume of data. To address this limitation, a directed acyclic graph (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 article, we propose a scalable and secure transaction attachment algorithm for the DAG-based blockchain IOTA. We determine two critical parameters through our experimental analysis: one for calculating the selection probability and the other for setting the threshold for abnormal transactions. First, we calculate the selection probability of unconfirmed transactions. Then, we select abnormal transactions whose selection probability falls below the predefined threshold to maintain the security. Finally, new transactions attach randomly to former transactions with a time computational complexity $O(n)$, ensuring the scalability. Through experiments comparing the proposed algorithm to the current transaction attaching algorithm, we demonstrate the scalability and security of our proposed algorithm.**

***Index Terms*—Distributed ledger system, Internet of Things (IoT), IOTA blockchain network, network modeling.**

I. Introduction

BLOCKCHAIN technology has garnered significant attention within both academic and industrial sectors for its innovative approach to decentralization [1], [2], [3]. For instance, there are practical blockchain use cases within the Internet of Things (IoT). Blockchain is implemented in transactive energy management (TEM) systems for IoT-enabled smart homes, achieving a 25% cost reduction through a privacy-preserving distributed algorithm that allows users to optimize energy usage [2]. Additionally, there is a consortium blockchain-based public integrity verification system (CBPIV) for the cloud storage in IoT, where auditor actions are tracked on the blockchain and monitored via smart contracts to ensure data integrity and security, alleviating computation demands on data owners [4]. With the increasing number of IoT devices introduces new demands on the scalability of blockchain networks [5]. Traditional chain-based blockchains, such as Bitcoin, face scalability bottlenecks due to their linear data structure, limiting their ability to handle the massive transaction volumes generated by IoT environments. In contrast, graph-based architectures offer a viable solution by inherently supporting parallel operations and make them a promising solution for overcoming the performance challenges posed by IoT systems.

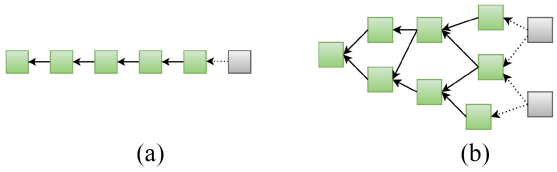


Fig. 1. Comparison of the blockchain data structure. (a) Chain. (b) DAG.

A novel blockchain data structure directed acyclic graph (DAG) is proposed to solve the scalability issue. As shown in Fig. 1, comparing to the chain, blockchain with DAG can processes transactions asynchronously. In these years, various DAG blockchains have been developed, such as IOTA [6], Byteball [7], Hashgraph [8], Fantom [9], 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 [6], [10]. Despite the novelty of IOTA 2.0, the consensus mechanism of IOTA 1.0 remains a representative and prototypical example of a DAG-based distributed ledger protocol. Although IOTA 1.0 involves the coordinator to ensure the legitimacy of transactions, which can be seen as a centralization factor, it still employs a validation process in which transactions must synchronize across all nodes in the network to reach distributed consensus. Each ledger entry needs to be validated, and IOTA nodes must consistently update their states to ensure alignment with other nodes. IOTA 1.0 is still being used in both research [11], [12] and general applications [13], [14], [15], [16], [17], and hence,

Received 1 October 2024; revised 11 November 2024 and 1 December 2024; accepted 15 December 2024. Date of publication 23 December 2024; date of current version 25 April 2025. This work was supported by Huawei. *(Corresponding author: Fengyang Guo.)*

Fengyang Guo is with the Munich Research Center, Huawei Technologies, 80992 Munich, Germany, and also with the Distributed Systems Group, TU Wien, 1040 Vienna, Austria (e-mail: fengyang.guo@dsg.tuwien.ac.at).

Artur Hecker is with the Munich Research Center, Huawei Technologies, 80992 Munich, Germany (e-mail: artur.hecker@huawei.com).

Schahram Dustdar is with the Distributed Systems Group, TU Wien, 1040 Vienna, Austria, and also with ICREA, Universitat Pompeu Fabra, 08002 Barcelona, Spain (e-mail: dustdar@dsg.tuwien.ac.at).

Digital Object Identifier 10.1109/JIOT.2024.3521680

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

still deserves attention from the research and development communities. 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 tip selection algorithm (TSA). The original IOTA protocol employs the Markov Chain Monte Carlo (MCMC) algorithm as its TSA, which utilizes a weighted random walk to attach new transactions. A critical parameter α is used in the MCMC algorithm. A larger value of α leads the walker through the most weighted branch which can be against to parasite chain attacks, and these transactions with lower weights may be ignored. 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, refer to transactions that are not attached into the tangle through the prescribed TSA. Examples of such abnormal transactions include parasite chains and lazy tips. 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 to accomplish a better scalability, meanwhile keeping the security of the tangle. One such effort was proposed by Bu et al. in the form of G-IOTA [18]. This approach involves each new transaction referencing three previous messages. The same team later proposed E-IOTA [19], 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 [12], 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.

Our aim is to enhance scalability while maintaining security of the DAG-based consensus mechanism. There are two main challenges to achieve the goal as follows.

1) *Challenge 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. Therefore, selecting an appropriate α that is sensitive to abnormal tips and attack patterns is the first challenge of this study.
2) *Challenge 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.

In this article, we propose a secure uniform random tip selection (S-URTS) algorithm that addresses the aforementioned challenges and ensures the scalability and security of the tangle. First, we detect and select out abnormal tips with abnormal selection probabilities, then attach new transactions using uniform random tip selection (URTS), which selects the tip from set of all tips randomly [20]. 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.

1) We propose a novel TSA, which can maintain both scalability and security of a DAG-based blockchain.
2) We determine a proper α for our proposed algorithm based on statistical data from multiple repetitive experiments. This α is sensitive and can be used to detect abnormal tips in most tangle cases.
3) We set the baseline for the normal tip distribution and detection of the abnormal tips.
4) We demonstrate the properties of the proposed algorithm through various experiments. The proposed TSA S-URTS takes similar time with other TSAs, but S-URTS has less number of tips and could defend against to the parasite chain attack.

The remainder of this article is organized as following. We illustrate the related analysis about the TSA and attacks in Section II. Section III is about the basic knowledge used in the proposed algorithm. We describe the design of the proposed algorithm in Section IV. In Section V, we design the experiment to determine the critical parameters and test the proposed algorithm. Then, we analyze the experiment result in Section VI. We conclude the whole article in Section VII.

II. 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.

A. 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 [21]. 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 [20]. In another study by the same team, reported in [22], 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.

B. Tangle TSA Variants

There exist several works proposing various algorithms to stabilize the number of tips. In G-IOTA [19], 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 [18]. 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. Ferraro et al. [23] proposed a hybrid TSA 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 [12], which determines the α size based on the standard deviation of the cumulative weight (CW). 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. Chen et al. [24] proposed a time-division-based TSA, which quickly identifies two tips for an incoming transaction by sorting tip values within a time slot. This approach reduces transaction verification time and decreases the number of lazy and permanent tips; however, it does not address the issue of parasite chain attacks.

All the above TSA variants have better performance than MCMC in maintaining a stable and minimum number of tips, but security and scalability were not approved simultaneously.

C. Tangle Security

The most prevalent form of threat 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 [25]. 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 and Miri [26] proposed 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. [27] analyzed the behavior strategies of IOTA nodes using the 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.

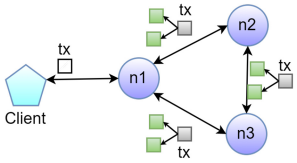


Fig. 2. Full and light node in IOTA network.

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

III. IOTA Preliminary

In this section, we present an overview of the fundamental concepts that underlie our work, encompassing IOTA-related concepts and the absorbing Markov chain. With respect to IOTA, we introduce the IOTA system, IOTA tangle, TSAs, and common attacks separately.

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.

A. IOTA System

The IOTA network is a distributed system that comprises two types of nodes: 1) full nodes (such as n_1–n_3) and 2) light nodes (clients), as shown in Fig. 2. A full node participates in the IOTA network by storing, exchanging, and synchronizing transaction data, which is eventually written into a local ledger called the tangle and organized as a DAG. A light node collects data from the user side and sends transactions to the IOTA network (a full node). For a full specification of an actual IOTA system (including node interactions, consensus, etc.), refer to [28].

B. 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.

In the tangle, each transaction possesses its own weight and a concept known as 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 transaction with lower CW values. The difference between the CW values of two connected transactions is referred to as the edge weight (EW).

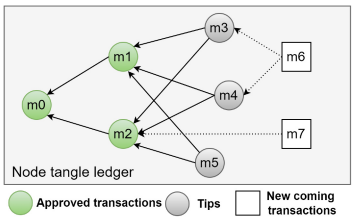


Fig. 3. Transaction attachment on the tangle.

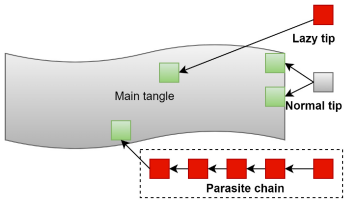


Fig. 4. Lazy tip and parasite chain.

C. IOTA Tip Selection Algorithm

IOTA attaches new incoming transactions to the tips through the TSA. 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 toward 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 CW, 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 common used TSA is the 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, IF recommends the use of MCMC for better security and stability of the tangle. URTS and unbiased random walk (URW) are theoretical TSAs and cannot be used in real-life implementation of DAG based DLT due to their vulnerability to parasite chain attacks [20].

Here we provide a detailed illustration of MCMC, as shown in Fig. 3, m_3–m_5 represent tips, while m_6 and m_7 denote new incoming transactions. A random walker walks from m_0 toward the end of the tangle. The transition probability between m_0 and m_1 is calculated using 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

$$p_{m_0 m_1} = \frac{e^{-\alpha EW_{m_0 m_1}}}{e^{-\alpha EW_{m_0 m_1}} + e^{-\alpha EW_{m_0 m_2}}}. \tag{1}$$

D. Attacks in IOTA

As noted in the IOTA whitepaper [6], the parasite chain attack is a primary threat to the IOTA tangle, with lazy tips as a specific variant. Our paper focuses on these two attacks as they pose significant security challenges to the IOTA consensus algorithm.

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 Fig. 3, transaction m_7 would be identified as a lazy tip, as it approves the already approved transaction m_2.

2) Parasite Chain: An attacker secretly constructs a subtangle that cites a transaction on the main tangle, thereby enhancing the CW of that transaction, as depicted in Fig. 4. The parasite chain can also be generated by a set of Sybil nodes. The concept of the parasite chain was first introduced by Popov [6]. Subsequent works [25], [26], [29], [30], [31], [32] published or recommended by the IF have extensively employed this type of parasite chain for security analysis and algorithm testing. This parasite chain exerts influence on the MCMC random walk process, directing the walker toward 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.

In practice, a hypothetical attacker could carry out a double-spending attack by attaching a parasitic chain to the tangle. As illustrated in the Fig. 4, the red squares denote transactions within the parasitic chain that conflict with an original transaction. The attacker waits for the confirmation of the original transaction before broadcasting the parasitic chain to the entire tangle, potentially validating the conflicting transaction [30].

The attacker's goal is to create a subtangle with a CW greater than the main tangle. If successful, new transactions would prefer to attach to the conflicting transaction. A parasitic chain is defined by the following parameters.

1) m is the length of the parasitic chain that references the main tangle.
2) λ represents the rate at which honest transactions are generated, which is related to the computing power of the honest network.
3) μ denotes the rate at which the attacker issues transactions on the parasitic chain, corresponding to the attacker's computing power.

It is possible to create a parasitic chain with a more complex structure. However, due to the complexity of the analysis, we will focus on a single-chain parasitic chain. Additionally, for the remainder of this article, we will assume that the honest majority assumption holds true [6]. This means that the computing power of the honest users is always greater than that of the attacker. This understanding is in line with the state-of-the-art in parasite chain prevention [25], [26], [29], [30], [31], [32].

E. Absorbing Markov Chain

An absorbing Markov chain is a special type of Markov chain that comprises two distinct states: 1) transient state and 2) absorbing state. At least one absorbing state is present in an absorbing Markov chain. Any transient state in an absorbing

TABLE I
VARIABLE DEFINITION

Variable	*Definition*
G_t	The DAG at time t
n	The number of transactions
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, 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, j
π_t	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
N	The size of the sub-tangle

Markov chain will inevitably reach an absorbing state with a probability of 1.

One important property of the absorbing Markov chain is the stationary distribution, which characterizes the distribution of all states after a sufficiently long period of time during which the distribution no longer undergoes any changes. In this context, the variable π represents a row vector of probabilities associated with the states. If π satisfies the property defined $\pi = \pi P$ (P is the transition probability of the absorbing Markov chain), it can be considered as the stationary distribution of the absorbing Markov Chain.

IV. ALGORITHM DESIGN

This section presents the proposed TSA S-URST. Before deploying the algorithm, we need to determine two important parameters: 1) random walk influence factor α and 2) threshold for abnormal tips 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 I.

A. Determine the α

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 article, we employ an experimental approach to determine the appropriate value for α. When the α value is too large, it amplifies the influence of varying weights on the probability distribution of tips, resulting in a more extreme distribution and making it challenging to identify anomalous transactions with lower weights. Conversely, when the α value is too small, it averages the probability distribution across different weights, reducing the sensitivity to abnormal transactions. Therefore, by testing several commonly used α values, it is possible to determine which value is most effective for distinguishing anomalies.

Algorithm 1 α Determination

Require: $set(\lambda)$, n, m, $set(\alpha)$
Ensure: α
for λ in $set(\lambda)$ **do**
  $G_\lambda = tangle\_generator(\lambda, n)$
end for
for i in [1,m] **do**
  $Pc_i = parasiteChain\_generator(\mathrm{i})$
end for
for G_λ 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_λ^m in $set(G_\lambda^m)$ **do**
  for α 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 α
Choose the α, whose mean is max and var is min.
return α

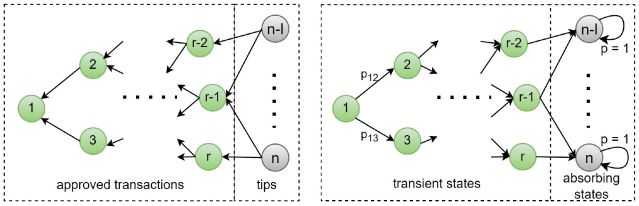
Algorithm 1 shows the whole process for α determination. First, we generate tangles for various values of λ using the URTS algorithm, and add parasite chains of varying lengths to the tangle. Subsequently, for each length of the parasite chain, the selection probability of both normal tips at the main 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.

B. Determine the 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 2. When determining the threshold value T, the moving average of a normal threshold over a sufficiently large sample set is used as the reference. If a value falls below this threshold, it is classified as anomalous.

Algorithm 2 Threshold Determination

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



(a) (b)

Fig. 5. Convert tangle to the absorbing Markov chain. (a) Tangle model. (b) Absorbing Markov chain.

This is because nodes generating abnormal transactions typically have lower computational power and are not integrated into the tangle following the system's normal procedures. Consequently, the calculated probability of such a node being selected will be lower than under normal circumstances. By calculating the selection probability distribution under typical conditions, the minimum moving average probability is determined, establishing a lower bound for normal selection probability. Transactions falling below this lower bound are classified as anomalies. 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.

C. Proposed TSA S-URTS

The present algorithm S-URTS commences by transforming the tangle into an absorbing Markov chain, followed by the computation of the probability distribution of all tips. Subsequently, the identification of the anomalous tip is carried out, and transactions are selected from the remaining tips. The primary steps involved in the algorithm are illustrated in Algorithm 3.

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 tangle. For example, the tangle shown in Fig. 5(a) includes n transactions, comprising r approved transactions and l tips. The Fig. 5(b) shows the absorbing Markov chain converted from that tangle in Fig. 5(a), which includes r transient states and l absorbing states with a transient probability of 1. The transient

Algorithm 3 Tip Selection

Require: $G(t)$, $V(t)$, $E(t)$, α, λ, 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

probability from state 1 to states 2 and 3 is p_{12} and p_{13}, respectively. If the number of transactions n in the tangle is bigger than the predefined subtangle size N, which is also the maximum random walk depth, then we will only calculate tips probability of the subtangle. The subtangle is a part of the tangle, constructed with the final tip to the N former transactions.

Then, we calculate the CW c_i of each transaction i and get the EW w_{ij} of each edge ij from (2). The affinity value between two states a_{ij} is influenced by α and calculate by (3). We obtain the transition probability p_{ij} for each pair of connected transactions from (4). After gathering this information, we construct the transition matrix P of the absorbing Markov chain, initiate the initial state π_0 as (5), calculate the stationary state distribution π to obtain the tip selection probability distribution D_t, through Fig. 6

$$w_{ij} = c_i - c_j \tag{2}$$

$$a_{ij} = \exp\left(-\alpha w_{ij}\right) \tag{3}$$

$$p_{ij} = \begin{cases} a_{ij} / \sum_{z \in N(i)} a_{iz}, & 1 \le i \le r \\ 1, & i = j,\ n - l \le i \le n \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

$$\pi_0 = [1, 0, \ldots, 0] \tag{5}$$

$$\pi_1 = \pi_0 P$$

$$\ldots$$

$$\pi = \pi_0 P^k. \tag{6}$$

At the end, we pick out the abnormal tips as shown in Fig. 6. We select the tips whose selection probabilities are below the threshold $T(t)$, and delete these abnormal tips from the tip set,

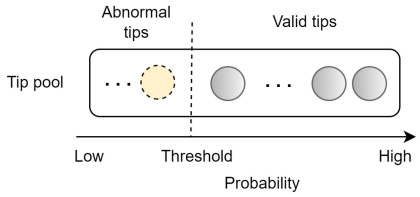


Fig. 6. Illustration of the abnormal tips selection.

TABLE II
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

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 $m_1, m_2, \ldots$ 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.

V. EXPERIMENT DESIGN

This section presents two experiments conducted for the proposed TSA: 1) experiments aimed at estimating the critical parameters of the algorithm and 2) experiments designed to evaluate the algorithm's performance.

A. Parameter Estimation

1) Determine α: The parameter α of the weighted random walk influences the transition probability between two connected transactions in the tangle. A small value of α results in a 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 experiments.

The experiment was conducted using varying values of λ and α. Some common values, including λ values of 5, 10, 15, and 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 Fig. 7. The box plots represent the selection probability distribution of the normal tips, where the box itself indicates the variability of the distribution. The orange line denotes the median value of this probability distribution. The blue points represent the abnormal tip selection probabilities, highlighting deviations from the expected range. 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 Fig. 8, 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.

2) 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, and 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. Fig. 9 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.

B. Algorithm Evaluation

The performance evaluation experiments comprise two aspects: 1) scalability and 2) 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.

1) Scalability: We compare the scalability of our proposed algorithm, S-URTS, with two other algorithms, namely, URTS

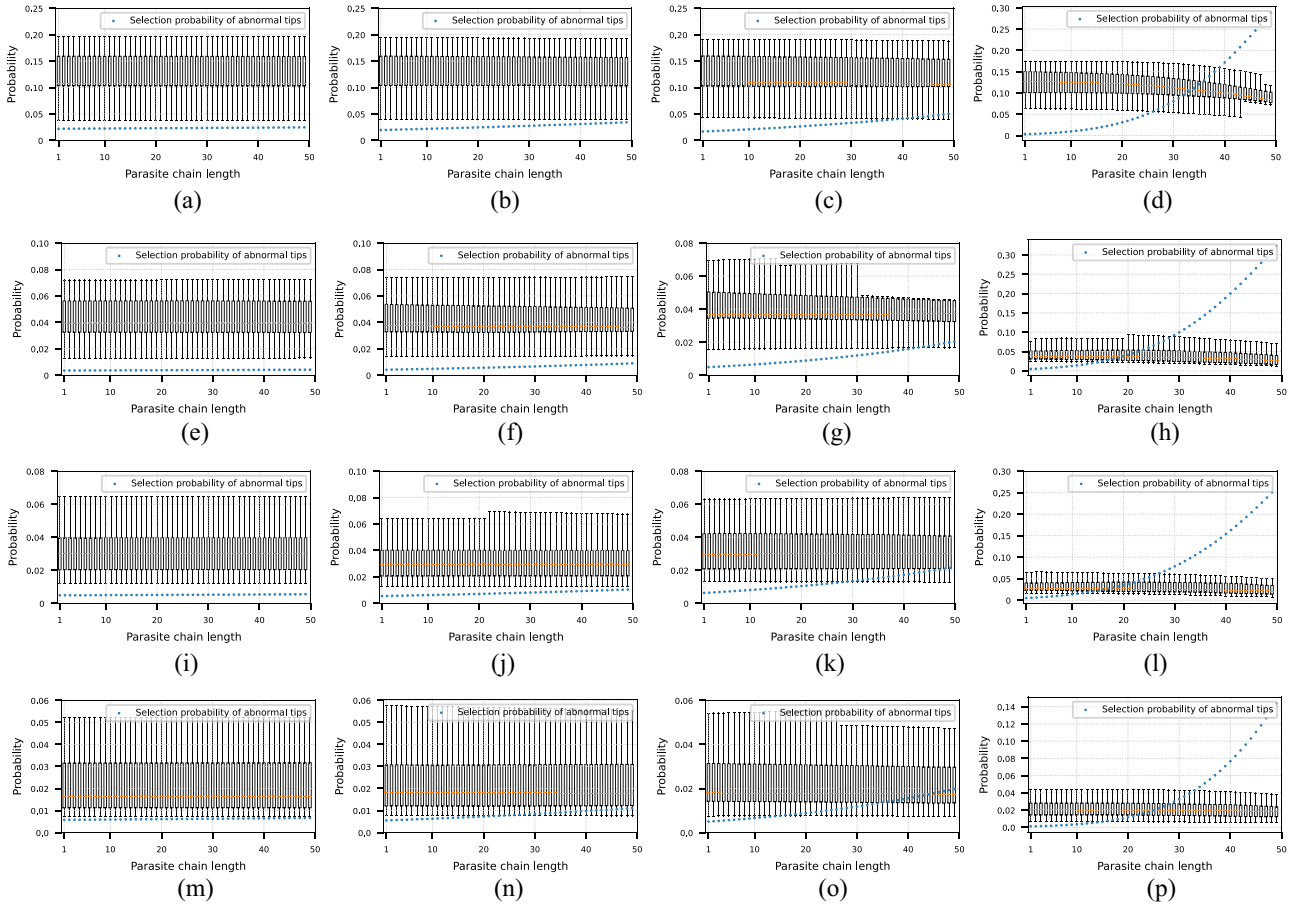


Fig. 7. Selection probability of the tip at the main tangle and at the parasite chain. (a) $\lambda = 5$ and $\alpha = 0.001$. (b) $\lambda = 5$ and $\alpha = 0.005$. (c) $\lambda = 5$ and $\alpha = 0.01$. (d) $\lambda = 5$ and $\alpha = 0.05$. (e) $\lambda = 10$ and $\alpha = 0.001$. (f) $\lambda = 10$ and $\alpha = 0.005$. (g) $\lambda = 10$ and $\alpha = 0.01$. (h) $\lambda = 10$ and $\alpha = 0.05$. (i) $\lambda = 15$ and $\alpha = 0.001$. (j) $\lambda = 15$ and $\alpha = 0.005$. (k) $\lambda = 15$ and $\alpha = 0.01$. (l) $\lambda = 15$ and $\alpha = 0.05$. (m) $\lambda = 20$ and $\alpha = 0.001$. (n) $\lambda = 20$ and $\alpha = 0.005$. (o) $\lambda = 20$ and $\alpha = 0.01$. (p) $\lambda = 20$ and $\alpha = 0.05$.

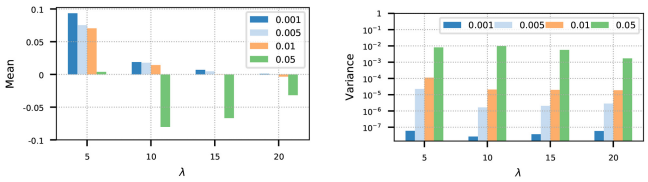


Fig. 8. Mean and variance of the probability difference between normal tip and parasite chain tip (with λ in 5, 10, 15, and 20, and α in 0.001, 0.005, 0.01, and 0.05). (a) Mean. (b) Variance.

TABLE III
EXPERIMENT SETUP: SCALABILITY

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

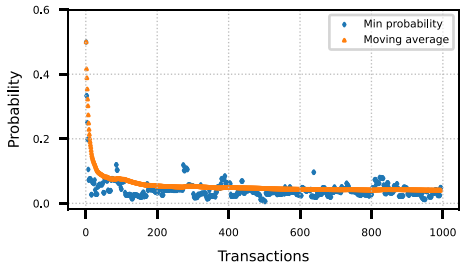


Fig. 9. Moving average of the minimum tip selection probability.

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 article, 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 III.

2) 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 subtangle 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

TABLE IV
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

be detected, as has been previously noted [25]. The detailed experimental settings are presented in Table IV.

VI. 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.

All experiments were conducted on a computer equipped with an Intel Core i5-8265U at 1.6 GHz CPU and 16 GB of RAM. Additionally, all algorithms were implemented in Python 3.8.

A. Scalability

We evaluate scalability using three parameters: 1) the number of tips; 2) consuming time for generating a new tangle; and 3) time complexity of the algorithm. When a large volume of new transactions enters the tangle, the network must process and validate these transactions. Each new transaction attaches to existing tips, completing their validation and becoming a new tip itself. A tangle with good scalability efficiently processes transactions, maintaining a stable number of tips, while poor scalability results in a backlog of unprocessed transactions, leading to an increasing number of tips. Consuming time for generating a new tangle is another key indicator—shorter consuming times reflect better scalability, while longer times indicate inefficiency. Reduced consuming time for attaching new transactions generally implies improved scalability, as the system can handle more transactions at a faster rate. This term refers to the assessment of an algorithm's computational complexity in relation to the time required for execution. A lower time complexity is generally indicative of improved scalability, as it allows the algorithm to handle an increasing number of transactions or operations more efficiently.

1) Number of Tips: The present study involves the analysis of tip counts during tangle generation using different TSAs, namely URST, MCMC1, MCMC5, and S-URTS. The raw data and the fitting line of the data of the number of tips are depicted in Fig. 10, which provides insights into the development trend of the number of tips with different TSAs

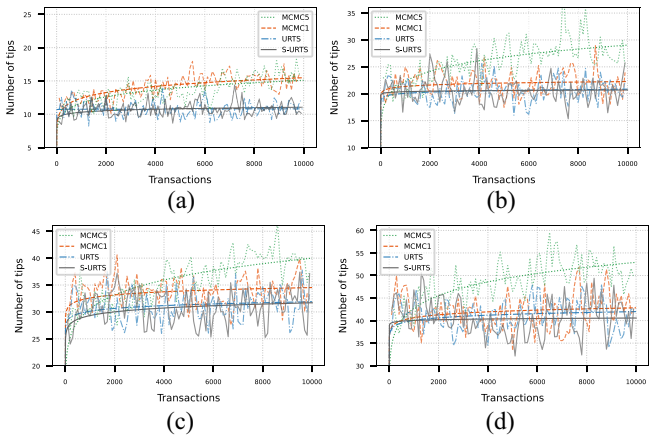


Fig. 10. Comparison of the number of tips development. (a) $\lambda = 5$. (b) $\lambda = 10$. (c) $\lambda = 15$. (d) $\lambda = 20$.

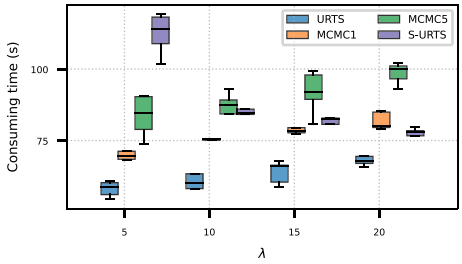


Fig. 11. Comparisons of consuming time.

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 10 000. 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 [6]. These experiments demonstrate that the number of tips of S-URTS can be maintained at a stable and low level.

2) Consuming Time: We collect the consuming time for generating the tangle with 10 000 transactions and show the results in Fig. 11.

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. 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

TABLE V
TIME COMPLEXITY

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

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.

3) Time Complexity: We conducted a comparative analysis of the time complexity of URTS, MCMC, and S-URTS for attaching new transactions.

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 CW of each transaction. Based on the definition of CW, the number of ancestors of each transaction must be calculated, resulting in a time complexity of $O(|V|^2)$. The subsequent step involves the computation of EW. The edge number is denoted as E, and the complexity of calculating EW 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 CW and transition probability, which are identical to those of the MCMC. The time 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 time complexity of the matrix calculation is $O(|V|^2/\lambda)$. For each round, the time 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 time complexity can be equivalent to $O((\lambda^2+1)|V|^3/\lambda^2+4|V|^2)$. When λ is large, the time complexity of the S-URTS algorithm is comparable to that of the MCMC algorithm.

Through our analysis of the number of tips, time required for computation and the time complexity, we have observed that for larger values of the parameter λ, the processing time of the S-URTS algorithm is shorter than that of both MCMC1 and MCMC5. Additionally, we have demonstrated that the S-URTS algorithm is capable of maintaining a stable and low number of tips while exhibiting similar time complexity to MCMC. These findings suggest that S-URTS exhibits better scalability compared to the aforementioned algorithms.

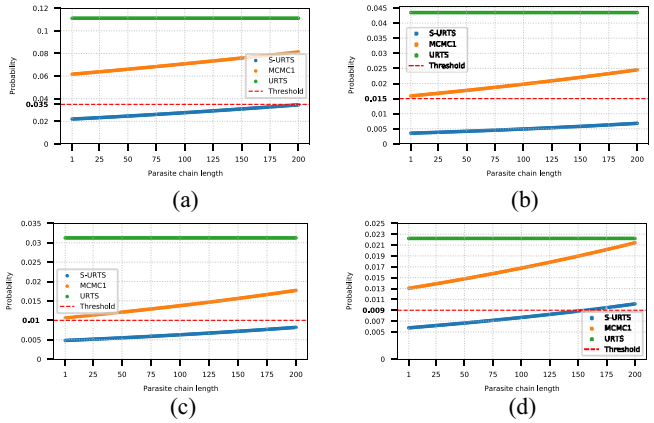


Fig. 12. Comparison of the selection probability of the parasite chain tip for various TSAs. (a) $\lambda = 5$. (b) $\lambda = 10$. (c) $\lambda = 15$. (d) $\lambda = 20$.

B. Security

To evaluate the security of the algorithm, we compare the tip selection probability of different TSAs. The probability of a tip being selected on a parasite chain can serve as a measure of the network's security. If the probability of a tip being selected on a parasite chain is higher than that on the main tangle, it indicates a vulnerability to attack. Conversely, if the selection probability on the parasite chain is lower, the network is more resistant to attacks. The tip selection probabilities on both the tangle and parasite chain are determined using different TSAs. A lower probability of tip selection on the parasite chain signifies a higher level of security. The lazy tip attack is a specific form of the parasite chain attack, particularly when the parasite chain consists of only a single transaction, which can then be considered a lazy tip attack. Therefore, our analysis focuses solely on the parasite chain attack, as it encompasses the lazy tip attack as well.

1) 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 Fig. 12.

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 S-URTS fall below the threshold T (red dashed line). 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.

VII. Future Work

This article 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: 1) node diversity; 2) network latency; and 3) security threats at the network layer.

A. 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.

B. 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.

C. 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.

VIII. Conclusion

This article presented an 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 [33] 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 [34], 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.

Acknowledgment

The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

References

[1] C. Zhang et al., "A blockchain-based model migration approach for secure and sustainable federated learning in IoT systems," *IEEE Internet Things J.*, vol. 10, no. 8, pp. 6574–6585, Apr. 2023.

[2] Q. Yang and H. Wang, "Privacy-preserving transactive energy management for IoT-aided smart homes via blockchain," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11463–11475, Jul. 2021.

[3] Z. Wang, Q. Chen, and L. Liu, "Permissioned blockchain-based secure and privacy-preserving data sharing protocol," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10698–10707, Jun. 2023.

[4] Y. Lin, J. Li, S. Kimura, Y. Yang, Y. Ji, and Y. Cao, "Consortium blockchain-based public integrity verification in cloud storage for IoT," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3978–3987, Mar. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9506885

[5] S. Zahra, W. Gong, H. A. Khattak, M. A. Shah, and H. Song, "Cross-domain security and interoperability in Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 11993–12000, Jul. 2022.

[6] S. Popov, "The tangle," IOTA Found., Geneva, Switzerland, White paper, 2016.

[7] A. Churyumov. "Byteball: A decentralized system for storage and transfer of value." 2016. [Online]. Available: https://byteball.org/Byteball.pdf

[8] L. Baird and A. Luykx, "The hashgraph protocol: Efficient asynchronous BFT for high-throughput distributed ledgers," in *Proc. Int. Conf. Omni-Layer Intell. Syst. (COINS)*, Aug. 2020, pp. 1–7.

[9] Q. Nguyen, A. Cronje, M. Kong, E. Lysenko, and A. Guzev, "Lachesis: Scalable asynchronous BFT on DAG streams," 2021, *arXiv:2108.01900*.

[10] S. Popov et al., *The Coordicide*, IOTA Found., Geneva, Switzerland, Jan. 2020, pp. 1–30.

[11] A. Tekeoglu, C.-F. Chiang, S. Sengupta, N. N. Ahmed, M. Stein, and D. Kusukuntla, "Optimized transaction processing in lightweight distributed ledger networks for Internet of Things," in *Proc. Int. Conf. Blockchain*, 2022, pp. 117–128.

[12] S. Rochman, J. E. Istiyanto, A. Dharmawan, V. Handika, and S. R. Purnama, "Optimization of tips selection on the IOTA tangle for securing blockchain-based IoT transactions," *Procedia Comput. Sci.*, vol. 216, pp. 230–236, Jan. 2023.

[13] S. Akbulut et al., "Designing a private and secure personal health records access management system: A solution based on IOTA distributed ledger technology," *Sensors*, vol. 23, no. 11, p. 5174, Jan. 2023.

[14] H. Zhang, M. Zaman, B. Stacey, and S. Sampalli, "A novel distributed ledger technology structure for wireless sensor networks based on IOTA tangle," *Electronics*, vol. 11, no. 15, p. 2403, Jan. 2022.

[15] M. Elhajj, H. Jradi, M. Chamoun, and A. Fadlallah, "LASII: Lightweight authentication scheme using IOTA in IoT platforms," in *Proc. 20th Mediterr. Commun. Comput. Netw. Conf. (MedComNet)*, Jun. 2022, pp. 74–83.

[16] N. R. Pradhan et al., "A blockchain based lightweight peer-to-peer energy trading framework for secured high throughput micro-transactions," *Sci. Rep.*, vol. 12, no. 1, Aug. 2022, Art. no. 14523.

[17] T. I. Meghla et al., "IOTA-based efficient and reliable scheme for Internet of Vehicles," in *Proc. Int. Conf. 4th Ind. Revolut. Beyond*, Singapore, 2022, pp. 385–400.

[18] G. Bu, W. Hana, and M. Potop-Butucaru, "E-IOTA: An efficient and fast metamorphism for IOTA," in *Proc. 2nd Conf. Blockchain Res. Appl. Innovat. Netw. Services (BRAINS)*, Sep. 2020, pp. 9–16.

[19] G. Bu, Ö. Gürcan, and M. Potop-Butucaru, "G-IOTA: Fair and confidence aware tangle," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 644–649.

[20] B. Kusmierz, W. Sanders, A. Penzkofer, A. Capossele, and A. Gal, "Properties of the tangle for uniform random and random walk tip selection," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, 2019, pp. 228–236.

[21] B. Kusmierz, P. Staupe, and A. Gal (IOTA Found., Geneva, Switzerland). *Extracting Tangle Properties in Continuous Time via Large-Scale Simulations*. Accessed: Aug. 23, 2018. 2018. [Online]. Available: https://https://www.iota.org/foundation/research-papers

[22] B. Kusmierz and A. Gal (IOTA Found., Geneva, Switzerland). *Probability of Being Left Behind and Probability of Becoming Permanent Tip in the Tangle*. Accessed: Apr. 16, 2018. 2018. [Online]. Available: https://https://www.iota.org/foundation/research-papers

[23] P. Ferraro, C. King, and R. Shorten, "On the stability of unverified transactions in a DAG-based distributed ledger," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3772–3783, Sep. 2020.

[24] Y. Chen, Y. Wang, B. Sun, and J. Liu, "Addressing the transaction validation issue in IOTA tangle: A tip selection algorithm based on time division," *Mathematics*, vol. 11, no. 19, p. 4116, 2023. [Online]. Available: https://www.mdpi.com/2227-7390/11/19/4116

[25] A. Penzkofer, B. Kusmierz, A. Capossele, W. Sanders, and O. Saa, "Parasite chain detection in the IOTA protocol," 2020, *arXiv:2004.13409*.

[26] S. Ghaffaripour and A. Miri, "Parasite chain attack detection in the IOTA network," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, May 2022, pp. 985–990.

[27] Y. Chen, Y. Guo, Y. Wang, and R. Bie, "Toward prevention of parasite chain attack in IOTA blockchain networks by using evolutionary game model," *Mathematics*, vol. 10, no. 7, p. 1108, Jan. 2022.

[28] F. Guo, X. Xiao, A. Hecker, and S. Dustdar, "Characterizing IOTA tangle with empirical data," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, 2020, pp. 1–6.

[29] V. Attias and Q. Bramas, "How to choose its parents in the tangle," in *Proc. 7th Int. Conf. Netw. Syst.*, 2019, pp. 275–280.

[30] A. Cullen, P. Ferraro, C. King, and R. Shorten, "Distributed ledger technology for IoT: Parasite chain attacks," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7112–7122, Aug. 2020.

[31] Y. Li et al., "Direct acyclic graph-based ledger for Internet of Things: Performance and security analysis," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1643–1656, Aug. 2020.

[32] M. Conti, G. Kumar, P. Nerurkar, R. Saha, and L. Vigneri, "A survey on security challenges and solutions in the IOTA," *J. Network Comput. Appl.*, vol. 203, Jul. 2022, Art. no. 103383.

[33] R. Nakanishi, Y. Zhang, M. Sasabe, and S. Kasahara, "IOTA-based access control framework for the Internet of Things," in *Proc. 2nd Conf. Blockchain Res. Appl. Innovat. Netw. Services (BRAINS)*, 2020, pp. 87–95. [Online]. Available: https://ieeexplore.ieee.org/document/9223293

[34] O. Cutajar, N. Moradpoor, and Z. Jaroucheh, "Using IOTA as an intervehicular trust mechanism in autonomous vehicles," in *Proc. 14th Int. Conf. Security Inf. Netw. (SIN)*, 2021, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/9699326