The ISC Int'l Journal of Information Security

July 2023, Volume 15, Number 2 (pp. 152–162)

http://www.isecure-journal.org

Analysis of IOTA Tangle Stability in High Transaction Rates

Habibollah Yajam [1], and MohammadAli Akhaee [1,*]

[1] *School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran*

ARTICLE INFO.

Article history:
Received: January 8, 2023
Revised: April 3, 2023
Accepted: July 1, 2023
Published Online: July 17, 2023

Keywords:
Blockchain, Consensus Algorithm, IoT, Scalability, Stability

Type: Research Article

doi: 10.22042/isecure.2023.380480.904

dor: 20.1001.1.20082045.2023.15.2.8.7

ABSTRACT

The future of the IoT requires new payment methods that can handle millions of transactions per second. IOTA cryptocurrency aims at providing such a solution. It uses a consensus algorithm based on directed acyclic graphs (DAG) called Tangle. A tip selection algorithm (TSA) is a part of Tangle that determines which unconfirmed blocks (tips) should be confirmed by new blocks. There is always a chance that a small number of valid blocks never get confirmed and become stale. If a significant part of the blocks becomes stale, the Tangle is considered unstable. In this paper, we mathematically prove that a TSA is stable at all transaction rates if and only if the probability of selecting all tips is at least $1/2n$ in which n is the total number of tips. Accordingly, we demonstrate that the current IOTA TSA would not be stable at high transaction rates.

1 Introduction

Soon after the birth of Bitcoin [1], the potential of blockchain and cryptocurrencies attracted the scientific research community. In its early years, scalability issues of Bitcoin's Nakamoto consensus became a research focus for many in academia and the industry. Thus, multiple works addressed this problem with different approaches [2–4]. One of the promising techniques is called directed acyclic graphs (DAG) which allows multiple blocks to get recorded and confirmed simultaneously. Thus, it can provide higher transaction throughput while maintaining the same level of decentralization [5–7].

In chain-based consensus algorithms, including Nakamoto consensus, blocks in the ledger form a tree graph. In such a graph, a directed edge from node A to node B indicates that block A contains the hash of block B, and therefore confirms it. However, blocks in a DAG-based ledger form a directed acyclic graph which means each block contains the hash of multiple blocks generated before it [8]. The first case of using a DAG for consensus algorithms is, to the best of our knowledge, "Inclusive." [9] While being studied well in the academic world, Inclusive never became a widely used scheme. About a year later, IOTA's Tangle was introduced [6], which was implemented later on. IOTA is probably the most widely-known DAG-based cryptocurrency. Its main goal is to provide fast and scalable payments for the Internet of Things (IoT), whereas other applications related to IoT are within the scope of its future [10].

One of the most critical challenges in consensus algorithms is handling conflicting transactions. When two or more transactions are individually valid but cannot be valid at once, we call them conflicting transactions or double-spends. Some DAG-based consensus algorithms handle these conflicting transactions by giving transactions an order such as SPECTRE [11] and PHANTOM [12]: Among all transactions that conflict with each other, the one that precedes the others is the only valid one. IOTA's Tangle [6] tries to solve this problem by staling all blocks that conflict except one. Thus, one of the conflicting transactions

* Corresponding author.
Email addresses: habib.yajam@gmail.com, akhaee@ut.ac.ir
ISSN: 2008-2045

receives a greater number of confirmations than the others.

Figure 1 shows a schematic view of Tangle's DAG. Since a hash pointer points to the previous block, the direction of edges is toward the root (genesis block), not the leaves (tips). In IOTA's Tangle, each block contains only one transaction, and it has precisely two parent blocks which are the two transactions it confirms. The algorithm selecting the parent transactions is called a tip selection algorithm. This algorithm has a fundamental role in Tangle's consensus algorithm since the participants use it to decide, among conflicting transactions, which one should gain more confirmations and replace the others accordingly [13]. IOTA's reference implementation currently employs an unweighted random walk tip selection algorithm, also known as the Markov Chain Monte Carlo (MCMC) algorithm (with α parameter equal to zero). This algorithm gives a higher probability of getting new confirmations to transactions that have already received some confirmations. Usually, when there is no malicious activity in the network, in IOTA's Tangle, the number of blocks without confirmations (also known as stale blocks) is expected to be insignificant. However, if the number of unconfirmed transactions gets larger and larger as the Tangle grows under certain conditions, we say that Tangle is unstable.

Our contributions in this paper can be summarized as follows:

- In Section 3 and Section 4, we use a mathematical model to evaluate the stability of the Tangle consensus algorithm and prove that any tip selection algorithm must assign all tip blocks at least a $1/2n$ probability of getting selected to guarantee stability in any transaction rate (n is the number of available tips).
- As a result of that theorem, in Section 5, we prove the stability of the uniform tip selection algorithm at any transaction rate. This confirms previous works that used totally different mathematical models [14].
- In Section 6, we show that the MCMC algorithm used in the current IOTA implementation cannot be stable at high transaction rates independent of its input parameters, due to the fact that it assigns some tips a probability of getting selected that is less than $1/2n$.
- We will discuss some solutions to fix this issue in Section 7.

2 Related works

Besides IOTA, few cryptocurrencies have similar DAG structures, such as Byteball [15], Nano [16], DagCoin [17], and Hedera Hashgraph [18].

Aside from the applied space, in academic research, multiple proposals have utilized a DAG-based structure to achieve higher transaction throughput and faster confirmation times. One of the most well-known examples is GHOST [19], which got implemented into Ethereum [20] with slight modifications later on. While GHOST employs DAGs, it is only an alternative method of choosing the main chain. Therefore, it ignores transactions outside the main chain.

To the best of our knowledge, block-DAGs were first introduced in Inclusive [9] to increase throughput by integrating transactions in off-the-main-chain blocks into the ledger. SPECTRE [11] employs transaction confirmation via a recursive election. The core of SPECTRE is a voting procedure that provides order between any pair of blocks. Votes are based on the location of the corresponding blocks. The authors show that the voting result quickly becomes irreversible and yields a consistent set of transactions. Another block-DAG consensus algorithm, PHANTOM [12], finds a cluster of well-connected blocks in the DAG and favors blocks inside that cluster while penalizing blocks outside of it. PHANTOM generalizes natural partial order in DAGs to total topological order.

GraphChain [21] aims at providing a more egalitarian approach to mining by proposing a method that allows low-powered miners to contribute to the networks' security and have steady incomes without joining mining pools. The authors also claim that GraphChain provides fast confirmations and scales with variations in transaction rates. Conflux [7] proposes an approach somewhat similar to Inclusive and GHOST. It also adds multiple improvements in the network layer and implements the results on the Bitcoin core.

Closer to our research topic, there have been a few research works on the analysis of the security and performance of Tangle. A work by Popov *et al.* [22] considers the circumstances in which each player attempts to optimize his attachment strategy. It also proves that under such conditions, Nash equilibrium exists. The authors show that selfish players prefer to join the proposed attachment strategy since their transactions will be less likely to be approved if they act otherwise. Another work by Barams [14] presents a formal analysis of the average number of unconfirmed transactions and the average confirmation time of a transaction in a discrete model. Subsequently, the paper provides proof that maintaining an honest majority in the network is necessary for protecting Tangle against attacks. A research paper by Kusmierz [23] deals with the simulation of the discrete-time model. Their work uses simulations to demonstrate the stability of the uniform tip selection algorithm and the risks of MCMC instability. This is done with high α

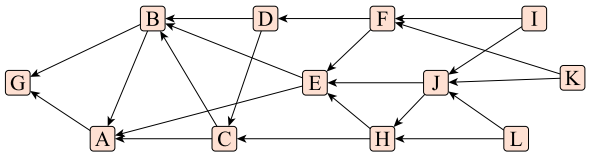


Figure 1. A schematic view of Tangle's DAG

values as the input parameter. In contrast, our work uses theoretical analysis that leads to an important theorem that determines the conditions a tip selection algorithm must have to be stable for any transaction rate.

The work of Kusmierz *et al.* [23] studies simulations of the continuous model. It verifies the Tangle whitepaper predictions. Furthermore, they explore the novel concept of a non-constant number of directly approved transactions k. As a final result, they have provided an analytical formula for the average number of tips in the discrete model, considering a uniform tip selection algorithm.

The research work of Cullen *et al.* [24] investigated the security of the IOTA Tangle under a particular type of attack scenario called double-spending. They have used a Markov Chain model to evaluate the resilience of the MCMC tip selection algorithm against malicious parties. They validated the results using simulations of random walks on randomly generated instances of Tangle.

Another related research work done by Ferraro *et al.* [25] uses a fluid model to describe tip selection dynamics. Utilizing a newly developed definition called "tip age," the authors show that the MCMC tip selection algorithm is unstable for a sufficiently large value of α. Also, using a uniform tip selection algorithm, all tips will eventually be approved under certain conditions. Their results are consistent with our research. However, our analysis extracts the necessary and sufficient conditions to make a tip selection algorithm stable for all block generation rates using a more general mathematical model. Additionally, we prove that there exists a block generation rate that makes the MCMC tip selection algorithm unstable even at the lowest possible value of α.

3 Preliminaries and Definitions

In IOTA's Tangle, each newly created block confirms exactly two previous blocks. These two blocks must be *local tips* which means that, in the view of the participant generating the transaction, they do not have any children. A participant in the network, due to delays in communications, sees a sub-DAG of all generated blocks. We call this sub-DAG the *local DAG*. While a block in one participant's local DAG could be a tip, that same block might not be a tip in another participant's local DAG. This could result in multiple confirmations for a single block from different participants.

When generating a new transaction, the choice of the two local tips for confirmation is based on an algorithm called the tip selection algorithm. While this algorithm is not forced by the protocol and participants are free to use their algorithm, it is shown that rational participants prefer using the same algorithm as most others [22]. One example of tip selection is the uniform tip selection algorithm. By this algorithm, two blocks are chosen with equal probability from the set of all local tips.

In the original whitepaper of IOTA [6], a tip selection algorithm called the MCMC algorithm is introduced. In this algorithm, a fixed weight is assigned to each block. The cumulative weight of a block is the sum of its descendants' weights in the local DAG plus its own. For example, as depicted in Figure 3, the accumulated weight of the point B is the sum of the weight of B itself and the accumulated weights of C, D, and E. The accumulated weight of childless blocks like E and D is equal to their weight.

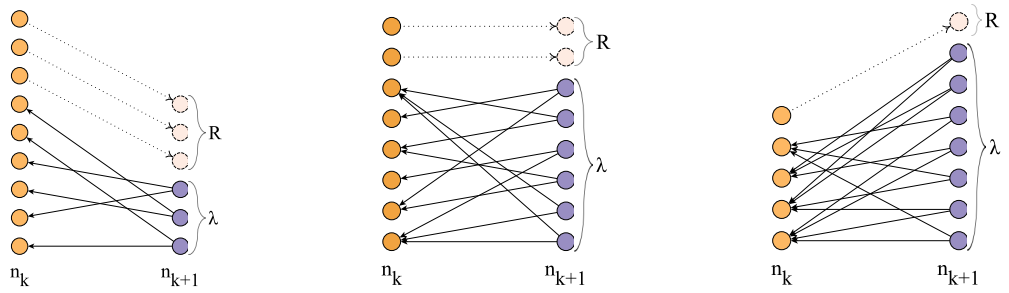
In the MCMC algorithm, to select a local tip, the algorithm assumes a virtual particle that travels through the edges of the local DAG. This particle travels from the genesis block, or a block far in the past, toward the tips. At each step, the particle moves from one block to one of its chosen children. The child is randomly selected with a probability derived from its cumulative weight. This travel continues until the particle lands on a block with no children. This block is what has been chosen.

When the virtual particle travels in the local DAG, the probability of selecting each child of a block is p_i for each child i and is calculated by Equation 1. W_x is the cumulative weight of block x. Currently, the weight of each block in IOTA's reference implementation is fixed and equal to 1.

$$p_i = \frac{e^{-\alpha(W_B - W_i)}}{\sum_{i \in children(B)} e^{-\alpha(W_B - W_i)}} \tag{1}$$

The value of α is a parameter that can be adjusted for increasing or decreasing the inclination toward selecting children with more confirmations. The lowest α value is zero. IOTA's whitepaper proposed 1 as an example value for α. In the initial version of IOTA's full-node Reference Implementation (IRI), α was defaulted to $\ln(3) \approx 1.1$ and later it changed to 0.001 [26]. In the 1.8.2 version released in October 2019, the default value of α is set to zero [27].

Any tip selection algorithm could be modeled as a random function $\Pi(.)$ that receives a local DAG D and



(a) Decreasing ($\delta_k < 0$) (b) Stable ($\delta_k = 0$) (c) Increasing ($\delta_k > 0$)

Figure 2. Difference in number of tips between consecutive rounds for various values of δ

assigns a probability vector to the set of all available local tips $\boldsymbol{P} = \Pi(D)$. Each element of this probability vector determines the probability of selecting the corresponding tip $\boldsymbol{P} = [P_1, P_2, \ldots, P_i, \ldots, P_{n_k}]$ which assigns the probability P_i to the tip t_i. For example, in a uniform tip selection algorithm, P_i is equal to $\frac{1}{n_k}$ for all tips.

3.1 Mathematical Model

In the rest of this paper, we will use the notations provided in Table 1. In the analysis of IOTA's Tangle stability, we assume a synchronous network model (SNM) [28]. Similar approaches have been used in the literature for analyzing Bitcoin's Nakamoto consensus [29] and Ethereum's GHOST protocol [30].

In this model, all the network participants are considered honest, and they share the same algorithm for tip selection, albeit with different randomness. The network starts with an initial state; then, participants proceed in lockstep, doing rounds. At the start of each round, all participants have the same view of the network (the DAG). In the kth round, network participants generate λ_k blocks independently, and each node uses a tip selection algorithm for choosing the parent blocks. We call λ_k the block generation rate of round k. Since in IOTA, each block contains exactly one transaction; there is no difference between the transaction generation rate and block generation rate. In the next round, the $k+1$ round, all blocks are delivered to all participants; thus, their view of the DAG is synchronized. As an assumption, for all k, the value of λ_k is less than an arbitrarily large value λ_{max}.

$$\forall k \in \mathbb{N}, \lambda_k < \lambda_{max} \tag{2}$$

3.2 Definitions

Definition 1. Difference in the number of tips. The value δ_k is the number of tips in the round $k+1$ subtracted by the number of tips in the round k:

$$\delta_k = n_{k+1} - n_k \tag{3}$$

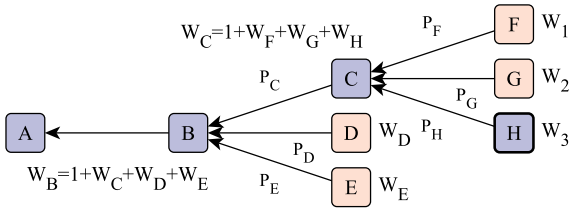


Figure 3. Calculation of accumulative weights in the MCMC algorithm

As shown in Figure 2, δ_k can be a negative or positive value under various conditions. The number of tips in each round could be deconstructed into two parts; the number of newly generated blocks which is λ_k, and the number of tips that remained from the last round (R_k):

$$\delta_k = \lambda_k + R_k - n_k \tag{4}$$

All tip selection algorithms are probabilistic. Accordingly, we normally use the expected value of δ_k. At the start of each round, all participants share the same view of the network and run the same tip selection algorithm. Therefore, the probability of choosing a tip i for all participants is equal to P_i. Subsequently, the probability that a tip does not get any confirmation from all λ_k new transactions that independently create $2\lambda_k$ confirmations is $(1-P_i)^{2\lambda_k}$. Throughout our analysis, we frequently use the expected value of δ_k conditional to n_k:

$$\begin{aligned}\mathbb{E}[\delta_k | n_k] &= \lambda_k + \mathbb{E}[R_k] - n_k \\ &= \lambda_k + \sum_{i=1}^{n_k} (1-P_i)^{2\lambda_k} - n_k\end{aligned} \tag{5}$$

Definition 2. Stable tip selection algorithm. A *tip selection algorithm* is *stable* if for any transaction generation rate $\lambda_k \in \mathbb{N}$ such that $\forall k \in \mathbb{N}, \lambda_k < \lambda_{max}$, the expected number of tips $\mathbb{E}[n_k]$ is a finite value:

$$\mathbb{E}[n_k] < \infty, \forall k \in \mathbb{N} \tag{6}$$

4 Theorems

Theorem 1 (Finite Number of Tips)**.** *If the expected number of tips is finite, then the number of tips is*

Table 1. Notations used in the manuscript

Sign	Description
T_k	Set of all available local tips in round k
n_k	Number of tips in round k ($n_k = \|T_k\|$)
δ_k	Difference between n_{k+1} and n_k
λ_k	Number of newly generated blocks in round k
P_i	Probability of selecting tip t_i from set of tips T
R_k	Tips remained from previous rounds in round k
$\boldsymbol{P}$	Probability vector $[P_1, P_2, \dots, P_n]$
$\mathbf{1}$	Vector of all ones with length n
P_{min}	Minimum value of the probability vector $\boldsymbol{P}$
$M_p(.)$	Power mean function with power p
a_k	Normalized value of λ_k over n_k ($\frac{\lambda_k}{n_k}$)

almost surely (a.s.) finite.

$$\mathbb{E}[n_k] < \infty \implies \Pr[n_k < \infty] = 1 \tag{7}$$

The proof of this theorem is out of the scope of this paper.

Lemma 1 (Boundedness of δ_k)**.** *The value of δ_k, difference in number of tips of the two consecutive rounds, is always bounded:*

$$\lambda_k - n_k \le \delta_k \le \lambda_k \tag{8}$$

Proof. The minimum value for the remaining tips is zero and the maximum is n.

$$0 \le R_k \le n_k \tag{9}$$
$$\lambda_k - n_k \le \lambda_k + R_k - n_k \le \lambda_k \tag{10}$$
$$\lambda_k - n_k \le \delta_k \le \lambda_k \tag{11}$$

As a result of Lemma 1 and Equation 2, we can say the value of δ_k is bounded for all k:

$$\forall k \in \mathbb{N} : \delta_k \le \lambda_k < \lambda_{max} \tag{12}$$

□

Theorem 2 (Stable Tip Selection Algorithms)**.** *A tip selection algorithm used by all participants of the network is stable if and only if:*

$$\forall k \in \mathbb{N}, \exists L \in \mathbb{R} \text{ s.t. if } n_k > L \text{ then } \mathbb{E}[\delta_k] n_k < 0 \tag{13}$$

The proof for Theorem 2 is presented in Appendix A.

Theorem 3 (Stability Conditions for Tip Selection Algorithms)**.** *A tip selection algorithm that creates the tip selection probability vector of $\boldsymbol{P}$ is stable in rate λ_k if and only if:*

$$\forall k \in \mathbb{N}, \exists L \in \mathbb{R} \text{ s.t. if } n_k > L \text{ then}$$
$$M_{2\lambda_k}(\mathbf{1} - \boldsymbol{P}) < \sqrt[2\lambda_k]{\frac{n_k - \lambda_k}{n_k}} \tag{14}$$

In this equation, $M_{2\lambda_k}$ is the power mean function with the exponent value of $2\lambda_k$.

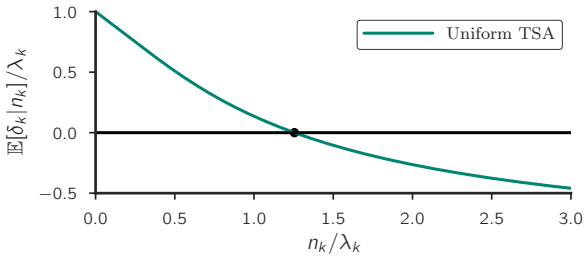


Figure 4. Theoretical expected value of δ_k against the number of tips n_k for uniform tip selection algorithm (both values are normalized by block generation rate λ_k)

The proof of Theorem 3 is in Appendix B.

Corollary 1. *If a tip selection algorithm is stable in the block generation rate λ_k, then it is also stable for any other rate less than λ_k. Consequently, if a tip selection algorithm is not stable for the block generation λ_k it is not stable for any rate higher than λ_k.*

The proof of Corollary 1 is presented in Appendix C.

Theorem 4. *A tip selection algorithm with the minimum tip selection probability $P_{min} = min(\boldsymbol{P})$ is stable for any transaction generation rate if and only if:*

$$\forall k \in \mathbb{N} : P_{min} > \frac{1}{2n_k} \tag{15}$$

Appendix D contains the proof for Theorem 4.

5 Stability of Uniform Tip Selection Algorithm

Using Theorem 3 and the fact that for a uniform tip selection algorithm $\forall P \in \boldsymbol{P} : P = \frac{1}{n_k}$ we have:

$$M_{2\lambda_k}(\mathbf{1} - \frac{1}{n_k}) < \sqrt[2\lambda_k]{\frac{n_k - \lambda_k}{n_k}} \tag{16}$$
$$\sum_{i=1}^{n_k} (1 - \frac{1}{n_k})^{2\lambda_k} < n_k - \lambda_k \tag{17}$$

Since the aim is to determine if Tangle is stable when the number of tips goes towards infinity, the value of n_k is large enough to utilize exponential approximation.

$$n_k e^{-2\frac{\lambda_k}{n_k}} < n_k - \lambda_k \tag{18}$$

Similar to the proof of Theorem 4, we assume $a_k = \frac{\lambda_k}{n_k}$ and use Lambert W function to deduce the following inequality:

$$a_k < 1 + a_k e^{\frac{-2}{a_k}}$$
$$< \frac{2}{W(\frac{-2}{e^2}) + 2} \approx 0.79681213 \tag{19}$$

Thus, for a uniform tip selection algorithm, if the ratio between the expected number of tips $\mathbb{E}[n_k]$ and the block generation rate λ_k is larger than $\frac{1}{0.7968} = 1.255$, then the expected value of δ_k is less than zero. Figure 4 shows the expected value of δ_k against n_k both

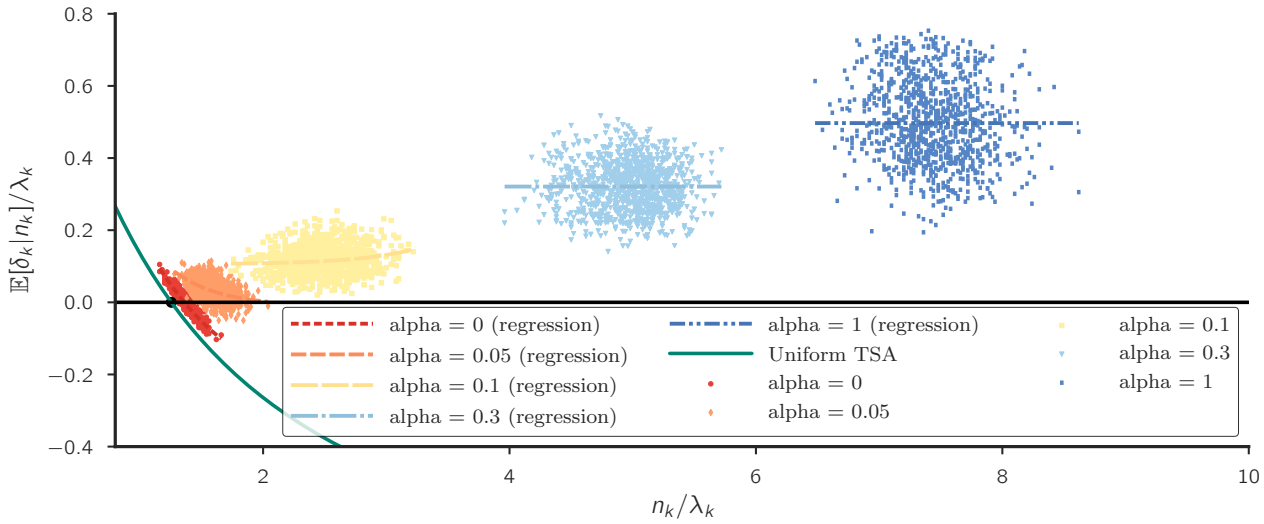


Figure 5. Normalazed expected value of δ_k conditioned to n_k for various values of α with a fixed block generation rate of 50

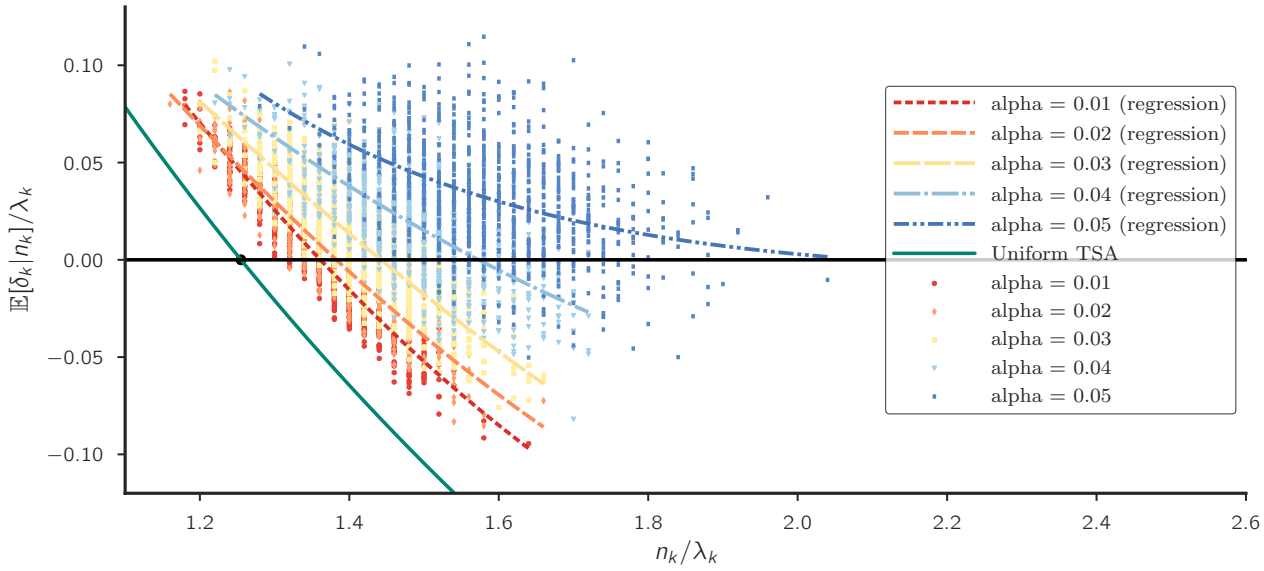


Figure 6. δ_k for various values of α while the block generation rate is fixed and equal to 50 (The graph shows that slight increase in the value of α steers the Tangle toward instability)

normalized by λ_k. When there are only small variations in the number of newly generated blocks in each round λ_k, the number of tips is likely to be approximately equal to $1.255\lambda_k$. This value acts as a stability point since values higher than that lead to negative expected growth in the number of tips, and values lower cause positive growth. On that account, the number of tips in each round is expected to be 25.5% more than the number of newly generated blocks. Similar results have been reported in [14] using a different mathematical model. Our model demonstrates this value as a stability point.

6 Analyzing the Stability of MCMC Algorithm

To find the probability that the MCMC algorithm assigns to each tip, we use a program to run the algorithm multiple times and analyze the output probability vector $\boldsymbol{P}$.

It is expected that different values of α in the MCMC tip selection algorithm will lead to a variety of stability levels. Lower α values result in probability distributions closer to uniform. Higher values of α cause more variance in the values of $\boldsymbol{P}$ elements.

Based on Theorem 4, we demonstrate that the

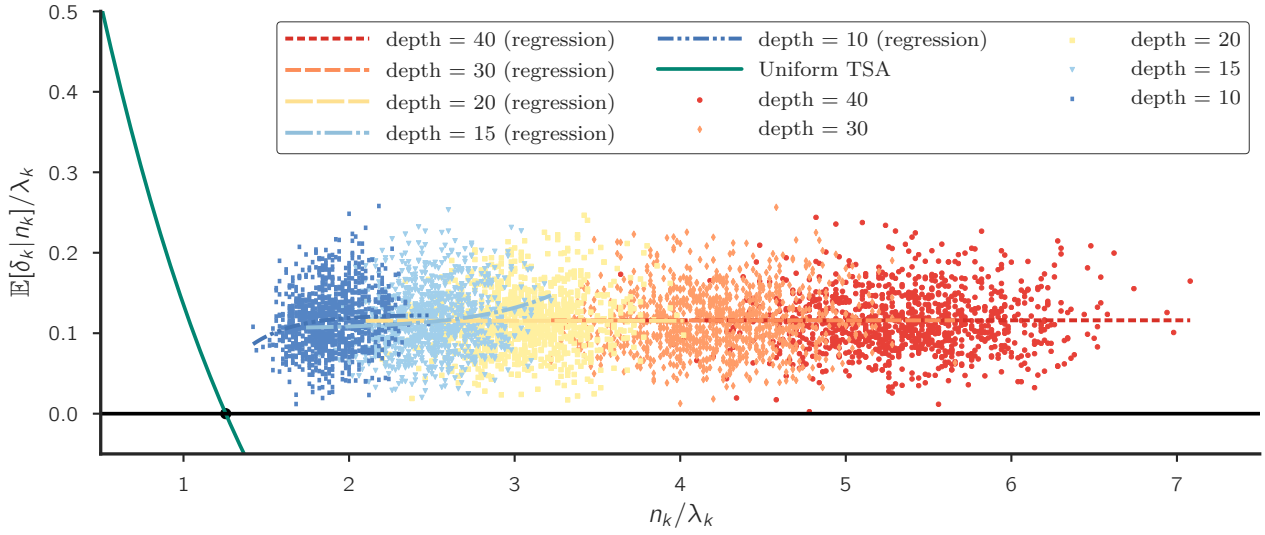


Figure 7. Normalized expected values of δ_k conditioned to n_k for $\alpha = 0.1$ calculated after various number of rounds

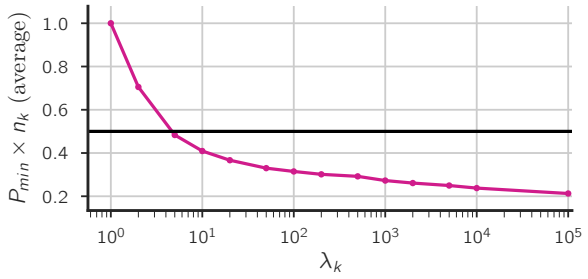


Figure 8. Value of P_{min} against different values of block generation rates λ_k for $\alpha = 0$ (Each value is extracted from more than 1000 executions)

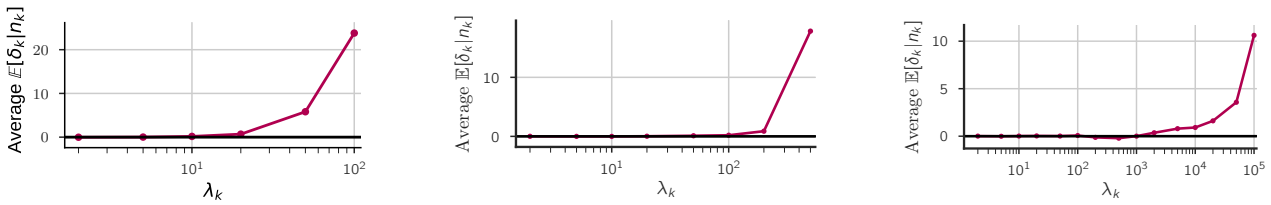
MCMC algorithm does not have the requirement for a stable tip selection algorithm at high transaction rates. To run the MCMC algorithm multiple times, we developed a program in the Python programming language using the "networkx" package. Our program runs the MCMC algorithm for various values of parameter α in any desirable block generation rate. It starts with a genesis block and adds new blocks to the DAG structure based on MCMC TSA. In each round, new blocks choose their parents by executing the MCMC tip selection algorithm with a preconfigured α parameter. After a certain number of rounds, we calculate the output probability vector $\boldsymbol{P}$ assigned to the tips.

Our first result using the program is depicted in Figure 5, which shows that for relatively high values of α, the number of stale blocks grows very fast. This figure is generated by calculating the value of $\mathbb{E}[\delta_k|n_k]/\lambda_k$ after 15 rounds of adding new blocks ($k = 15$). In each round, the block generation rate is 50. For each α value, more than 1000 experiments with different random values are executed. We used an exponential regression for each α value to better illustrate the results. The results show that for higher values of α, the Tangle is extremely unstable since there is a significant growth in the number of tips. The instability of MCMC tip selection with large values of α has been reported in [14, 25]. In Figure 5, the result from the theoretical assessment of the uniform tips selection algorithm is also depicted. This shows that the uniform tip selection algorithm has the lowest expected number of tips at all block generation rates.

Figure 6 shows another result of running the MCMC algorithm. It provides similar information to the previous figure but with very small variations in the value of α. This figure emphasizes that every slight change in α could significantly affect the Tangle's stability. These results are for a block generation rate of 50 new blocks per round. The results of the expected number of tips are calculated after running 15 rounds.

Results shown in Figure 7 confirm that when the average of $\mathbb{E}[\delta_k|n_k]$ is above zero, rounds of running MCMC lead to a higher number of tips in the next rounds (instability). The figure shows the number of tips for a block generation rate of 50 and $\alpha = 0.1$ with varying values of rounds (depth).

Figure 8 shows that the MCMC algorithm assigns probabilities less than $\frac{1}{2n_k}$ to some tips even when the block generation rate is low. According to Theorem 4, if the minimum probability of choosing a tip among all available tips is less than $\frac{1}{2n_k}$, then there exists a block generation rate that makes the Tangle unstable. Results of execution of MCMC algorithm in Figure 8 indicates that the value of P_{min} for MCMC algorithm with $\alpha = 0$ is less than $\frac{1}{2n_k}$ for most block generation rates and it drops lower by increasing the rate.



(a) $\alpha = 0.1$ (b) $\alpha = 0.01$ (c) $\alpha = 0$

Figure 9. Average value of $\mathbb{E}[\delta_k|n_k]$ for different values of α (Figures are depicted in block generation rate λ_k intervals from 1 up to the point that the Tangle becomes unstable. Each data point is extracted from more than 1000 executions)

In Figure 9 the value of $\mathbb{E}[\delta_k|n_k]$ for different values of α is depicted. These results show that for any value of α in the MCMC tip selection algorithm, there exists a block generation rate that makes IOTA's Tangle unstable (positive $\mathbb{E}[\delta_k|n_k]$ value). Although this point is relatively high for very small values of α (e.g., zero), still a rate that leads to Tangle instability exists. This is consistent with our theoretical results that the MCMC algorithm cannot be stable when the block generation rate (λ_k) approaches infinity.

7 Discussion about Remedies

Based on the theoretical analysis in Section 4 and Section 6, the MCMC algorithm with $\alpha = 0$, which, at the time of writing, is being used for IOTA cryptocurrency cannot be stable for high transaction rates. As IoT networks handle millions of transactions per second, this limitation would eventually halt the growth of the cryptocurrency. Based on the results of our study, we suggest using different tip selection algorithms to fix this issue.

In Section 5, proofs for the stability of the uniform tip selection algorithm for any block generation rate have been presented. Other research works reported confirming results [14, 25, 31]. As a remedy to the stability problems in IOTA's Tangle, we suggest using a uniform tip selection algorithm instead of the MCMC algorithm. Although this approach affects the role of the tip selection algorithm against double-spending attacks, the designers could use other more studied techniques for tackling such attacks, such as the proof-of-stake voting approach of [32], which has gained much attention recently.

On the other hand, the protocol designers could increase the number of required parents for a block to be accepted in the Tangle to $m > 2$. This could change the condition of Theorem 4 to the following:

$$\forall k \in \mathbb{N} : P_{min} > \frac{1}{m \times n_k} \tag{20}$$

Although this does not lead to stability at all block generation rates, it could push the instability point to a much higher value.

One of the concerns regarding IOTA's Tangle is determining which transaction generation rates for different tip selection algorithms and network conditions are stable. This is especially relevant for users and developers who use MCMC TSA, as, according to our findings is not stable in high transaction rates. A possible future work could use a different mathematical framework and more extensive simulations to derive an upper bound for stable transaction generation rates when utilizing MCMC TSA. It could also compare it with other TSA. This would help users and developers decide about IOTA for their IoT applications.

8 Conclusion

In this paper, we have presented a mathematical model to analyze the stability of Tangle, a DAG-based consensus algorithm for IOTA cryptocurrency. Our main findings are:

- We derived the necessary and sufficient conditions for stable tip selection algorithms. These conditions require that all tips have at least a $1/(2n)$ selection probability, where n is the number of tips.
- We proved the stability of the uniform tip selection algorithm, which selects tips randomly with equal probability, at all transaction rates. This confirms previous works that used different models.
- We showed that the Markov chain Monte Carlo tip selection algorithm, used in IOTA's implementation, is unstable at high transaction rates. We discussed the security and scalability implications of this result and suggested some solutions.

In future work, we aim to extend our analysis to other tip selection algorithms proposed in the literature. We will compare their performance and stability with the uniform tip selection algorithm.

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electic cash system, 2008.

[2] Ittay Eyal, Adem Efe Gencer, Emin Gun Sirer,

and Robbert Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 45–59, Santa Clara, CA, March 2016. USENIX Association.

[3] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, 2016.

[4] Ali Dorri, Salil S. Kanhere, Raja Jurdak, and Praveen Gauravaram. Lsb: A lightweight scalable blockchain for iot security and anonymity. *Journal of Parallel and Distributed Computing*, 134:180–197, 2019.

[5] Yonatan Sompolinsky and Aviv Zohar. Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 507–527. Springer, 2015.

[6] Serguei Popov. The tangle, 2016.

[7] Chenxing Li, Peilun Li, Dong Zhou, Wei Xu, Fan Long, and Andrew Yao. Scaling nakamoto consensus to thousands of transactions per second, 2018.

[8] Joanna Moubarak, Maroun Chamoun, and Eric Filiol. On distributed ledgers security and illegal uses. *Future Generation Computer Systems*, 113:183–195, 2020.

[9] Yoad Lewenberg, Yonatan Sompolinsky, and Aviv Zohar. Inclusive block chain protocols. In *International Conference on Financial Cryptography and Data Security*, pages 528–547. Springer, 2015.

[10] 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, Dec 2018.

[11] Yonatan Sompolinsky, Yoad Lewenberg, and Aviv Zohar. Spectre: A fast and scalable cryptocurrency protocol. Cryptology ePrint Archive, Report 2016/1159, 2016. https://eprint.iacr.org/2016/1159.

[12] Yonatan Sompolinsky and Aviv Zohar. Phantom: A scalable blockdag protocol. Cryptology ePrint Archive, Report 2018/104, 2018. https://eprint.iacr.org/2018/104.

[13] Wellington Fernandes Silvano and Roderval Marcelino. Iota tangle: A cryptocurrency to communicate internet-of-things data. *Future Generation Computer Systems*, 112:307–319, 2020.

[14] Quentin Bramas. The stability and the security of the tangle, 2018.

[15] Anton Churyumov. Byteball: A decentralized system for storage and transfer of value, 2016.

[16] Colin LeMahieu. Nano: A feeless distributed cryptocurrency network, 2018.

[17] Yary Ribero and Daniel Raissar. Dagcoin whitepaper, 2020.

[18] Leemon Baird. The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance. Technical Report Tech Reports SWIRLDS-TR-2016-01, Swirlds, 2016.

[19] Yonatan Sompolinsky and Aviv Zohar. Accelerating bitcoin's transaction processing. fast money grows on trees, not chains. Cryptology ePrint Archive, Report 2013/881, 2013. https://eprint.iacr.org/2013/881.

[20] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[21] 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*, BCC '18, page 21–33, New York, NY, USA, 2018. Association for Computing Machinery.

[22] Serguei Popov, Olivia Saa, and Paulo Finardi. Equilibria in the tangle. *arXiv preprint arXiv:1712.05385*, 2017.

[23] Bartosz Kusmierz, Philip Staupe, and Alon Gal. Extracting tangle properties in continuous time via large-scale simulations, 2018.

[24] Andrew Cullen, Pietro Ferraro, Christopher King, and Robert Shorten. On the resilience of dag-based distributed ledgers in iot applications. *IEEE Internet of Things Journal*, 7(8):7112–7122, Aug 2020.

[25] P. Ferraro, C. King, and R. Shorten. On the stability of unverified transactions in a dag-based distributed ledger. *IEEE Transactions on Automatic Control*, 65(9):3772–3783, 2020.

[26] Alon Elmaliah. iotaledger/iri, 2020.

[27] Gal Rogozinski. iotaledger/iri, 2020.

[28] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

[29] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 281–310, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[30] Lei Yang, Vivek Bagaria, Gerui Wang, Mohammad Alizadeh, David Tse, Giulia Fanti, and Pramod Viswanath. Prism: Scaling bitcoin by 10,000x, 2020.

[31] 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.

[32] Vitalik Buterin, Diego Hernandez, Thor Kamphefner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Ying Wang, and Yan X Zhang. Combining ghost and casper, 2020.

A Proof of Theorem 2

Proof. Proof of necessity. For a stable tip selection algorithm, if the expected number of tips is finite for all k then there exists a value $L^* \in \mathbb{R}$ greater than all values of $\mathbb{E}[n_k]$.

$$\mathbb{E}[n_k] < L^* < \infty \tag{A.1}$$

Without loss of generality, we can replace k with $k+1$ and subtract n_k from both sides of the inequality:

$$\mathbb{E}[\delta_k | n_k] < L^* - n_k \tag{A.2}$$

Since $n_k > 0$, we can deduce that for values of n_k greater than L^*:

$$\mathbb{E}[\delta_k | n_k] < 0 \tag{A.3}$$

As a consequence, for a *stable* tip selection algorithm the condition is necessary. □

Proof. Proof of sufficiency. First:

$$\text{if } n_k > L \text{ then } \mathbb{E}[\delta_k | n_k] < 0 \implies \tag{A.4}$$
$$\text{if } n_k > L \text{ then } \mathbb{E}[n_{k+1}] < n_k \tag{A.5}$$

By replacing k with $k-1$, we have:

$$\text{if } n_{k-1} > L \textit{ then } \mathbb{E}[n_k] < n_{k-1} \tag{A.6}$$

From Lemma 1, we have:

$$\delta_k < \lambda_{max} \tag{A.7}$$

Thus, the expectation of δ_k conditional to any other variable is less than λ_{max}:

$$\mathbb{E}[\delta_k | n_k] < \lambda_{max} \tag{A.8}$$
$$\mathbb{E}[n_{k+1}] < \lambda_{max} + n_k \tag{A.9}$$

We can simply replace k with $k-1$:

$$\mathbb{E}[n_k] < \lambda_{max} + n_{k-1} \tag{A.10}$$

From Equation A.6 and Equation A.10:

$$\mathbb{E}[n_k] < \max(n_{k-1}, \lambda_{max} + L) \tag{A.11}$$

It is safe to assume that the number of tips in the initial state is finite.

$$\exists L > 0 : n_0 < L_0 \tag{A.12}$$

Using induction from Equation A.11 and Equation A.12:

$$\mathbb{E}[n_k] < \max(L_0, \lambda_{max} + L) \implies$$
$$\mathbb{E}[n_k] < L^{'} < \infty \tag{A.13}$$

□

B Proof of Theorem 3

Proof. From Theorem 2 and the definition in Equation 5, a tip selection algorithm is stable if and only if:

$$\text{if } n_k > L \text{ then } \lambda_k + \sum_{i=1}^{n_k} (1 - P_i)^{2\lambda_k} - n_k < 0 \tag{B.1}$$

$$\text{if } n_k > L \text{ then } \frac{1}{n_k} \sum_{i=1}^{n_k} (1 - P_i)^{2\lambda_k} < \frac{n_k - \lambda_k}{n_k} \tag{B.2}$$

Since for $n_k < \lambda_k$ definitely the number of tips will increase, we can assume that both sides of the inequality are positive values:

$$\text{if } n_k > L \text{ then } \sqrt[2\lambda_k]{\frac{1}{n_k} \sum_{i=1}^{n_k} (1 - P_i)^{2\lambda_k}} < \sqrt[2\lambda_k]{\frac{n_k - \lambda_k}{n_k}} \tag{B.3}$$

Using the definition of the power mean function:

$$\text{if } n_k > L \text{ then } M_{2\lambda_k}(\mathbf{1} - \boldsymbol{P}) < \sqrt[2\lambda_k]{\frac{n_k - \lambda_k}{n_k}} \tag{B.4}$$

□

C Proof of Corollary 1

Proof. From the definition of δ_k in Equation 5, it can be inferred that the second partial derivative of δ_k with respect to λ_k is non-negative for all $\lambda_k \geq 0$.

$$\forall \lambda_k \geq 0 : \frac{\partial^2 \delta_k}{\partial \lambda_k^2} \geq 0 \tag{C.1}$$

Also, the value of δ_k is zero for $\lambda_k = 0$ since when there exists no newly generated block, then the number of tips must not change.

$$for\ \lambda_k = 0 : n_k = 0 \tag{C.2}$$

if δ_k is less than zero for the block generation rate λ_k, then it is also less than zero for any rate less than λ_k. □

D Proof of Theorem 4

Proof. As a result of Corollary 1, if a tip selection algorithm is stable for $\lambda_k \to \infty$, then it is stable for all values of λ_k.

One of the properties of the power mean function is that when the exponent approaches infinity ($2\lambda_k \to \infty$), its output approaches the maximum value of the input vector.

$$\lim_{\lambda_k \to \infty} M_{2\lambda_k}(\mathbf{1} - \boldsymbol{P}) = \max(\mathbf{1} - \boldsymbol{P}) = 1 - P_{min} \tag{D.1}$$

Therefore, from Theorem 3, we can infer that if the following inequality holds for $\lambda_k \to \infty$, then for all values of $\lambda_k > 0$

$$1 - P_{min} < \sqrt[2\lambda_k]{\frac{n_k - \lambda_k}{n_k}} \tag{D.2}$$

Since all values are positive, we may raise both sides to the power of $2\lambda_k$.

$$(1 - P_{min})^{2\lambda_k} < \frac{n_k - \lambda_k}{n_k} \tag{D.3}$$

We know that n_k is at least as large as λ_k, thus P_{min} which is smaller than $\frac{1}{n_k}$ approaches zero. Therefore, we can safely use this exponential approximation:

$$e^{-2\lambda_k P_{min}} < \frac{n_k - \lambda_k}{n_k} \tag{D.4}$$

With a change of variables $a_k = \frac{\lambda_k}{n_k}$. We also assume that P_{min} is equal to $\frac{1}{n_k}$ multiplied by a constant coefficient θ. ($P_{min} = \frac{\theta}{n_k}$)

$$e^{-2\theta a_k} + a_k - 1 < 0 \tag{D.5}$$

This inequality can be solved with a close form formula using *Lambert W Function*.

$$2\theta > 1 \tag{D.6}$$

$$0 < a_k < \frac{1}{2\theta} W(-\frac{2\theta}{e^{2\theta}}) + 1 \tag{D.7}$$

Here, W is the Lambert W Function. From Equation D.6 it can be inferred that:

$$\theta > \frac{1}{2} \tag{D.8}$$

$$P_{min} > \frac{1}{2n_k} \tag{D.9}$$

From Equation D.7, we can calculate the lowest L that all values of n_k larger than L result in a negative value of $\mathbb{E}[\delta_k | n_k]$. This is the limit L explained in Theorem 2.

$$L = \frac{2\theta}{W\Big(-\frac{2\theta}{e^{2\theta}}\Big) + 2} \lambda_k \tag{D.10}$$

□

Habibollah Yajam is a Ph.D. student at the University of Tehran in the Secure Communication Lab. He graduated with his M.Sc. degree from the Sharif University of Technology in Telecommunication Engineering - Cryptography. His research background mostly concentrates on Blockchain Technology, Consensus Algorithms, Privacy-Preserving Protocols, and Anonymous Communications.

MohammadAli Akhaee received his B.Sc. degree in both electronics and communications engineering from the Amirkabir University of Technology, and the M.Sc. and Ph.D. degree in communication systems from the Sharif University of Technology in 2005 and 2009, respectively. He has awarded a governmental Endeavour research fellowship from Australia in 2010. He is an author/co-author of more than 40 papers and holds one Iranian patent. He served as the technical program chair of EUSIPCO'11. Dr. Akhaee serves as a faculty member at the College of Eng., University of Tehran, Tehran, Iran. His research interests include multimedia security, cryptography, watermarking, and statistical signal processing.