Parasite Chain Detection in the IOTA Protocol

Andreas Penzkofer
IOTA Foundation, Berlin, Germany

Bartosz Kusmierz
Department of Theoretical Physics, Wroclaw University of Science and Technology, Poland

Angelo Capossele
IOTA Foundation, Berlin, Germany

William Sanders
IOTA Foundation, Berlin, Germany

Olivia Saa
Department of Applied Mathematics, Institute of Mathematics and Statistics, University of São Paulo, Brazil

─── **Abstract** ───

In recent years several distributed ledger technologies based on directed acyclic graphs (DAGs) have appeared on the market. Similar to blockchain technologies, DAG-based systems aim to build an immutable ledger and are faced with security concerns regarding the irreversibility of the ledger state. However, due to their more complex nature and recent popularity, the study of adversarial actions has received little attention so far. In this paper we are concerned with a particular type of attack on the IOTA cryptocurrency, more specifically a *Parasite Chain* attack that attempts to revert the history stored in the DAG structure, also called the *Tangle*.

In order to improve the security of the Tangle, we present a detection mechanism for this type of attack. In this mechanism, we embrace the complexity of the DAG structure by sampling certain aspects of it, more particularly the distribution of the number of approvers. We initially describe models that predict the distribution that should be expected for a Tangle without any malicious actors. We then introduce metrics that compare this reference distribution with the measured distribution. Upon detection, measures can then be taken to render the attack unsuccessful. We show that due to a form of the Parasite Chain that is different from the main Tangle it is possible to detect certain types of malicious chains. We also show that although the attacker may change the structure of the Parasite Chain to avoid detection, this is done so at a significant cost since the attack is rendered less efficient.

1 Introduction

With the arrival of Bitcoin [24] a new decentralized payment system based on a trust-less peer-to-peer network has established. Bitcoin is essentially a protocol for reaching consensus between independent entities - that do not need to trust each other - on a chronologically ordered record of transactions. This data structure, which is also called a *blockchain*, is now a cornerstone of many other Distributed Ledger Technologies (**DLT**s) [6], [14], [17], [23].

Despite their great success, events of congestion and the correlated high transaction (**tx**) fees [15] show that limitations in throughput, which can become costly, exist for these types of DLTs. Since these events are effectively created due to the bottleneck of a limit of txs that can be processed, scalability has become a core research topic. Furthermore, this issue also hinders the adoption of the technology for applications such as Internet of Things (IoT) [12].

To overcome scalability issues, several techniques have been proposed ranging from increasing block size and frequency, to side chains [10], "layer-two" structures like Lightning Network [26], Sharding [22], and other consensus mechanisms [14, 17, 7]. Some DLT's have also replaced the blockchain structure with a directed acyclic graph (**DAG**). This approach is used in IOTA [27] and other protocols [19], [9], [5], [29], [4], [20], [8], [16]. DAG-based protocols reach consensus on a *partially* ordered log of transactions, allowing the log to have width and which can increase the throughput of the system.

In this paper, we study the DAG-based IOTA protocol introduced in [27], where transactions are recorded in a DAG dubbed the *Tangle*. The vertices in this DAG are transactions. If there is an edge between two transactions $x \leftarrow y$, we say that y (directly) approves x. If there is a directed path from y to x but no edge, y indirectly approves x. A transaction with no approvers is called a *tip*. Under the IOTA protocol, all incoming transactions attach themselves to the Tangle by approving two (not necessarily distinct) tips.

1.1   The IOTA protocol

Since the DAG structure is heavily determined by the order and manner in which txs are approved, the algorithm used to select the tips which a new transaction will attach itself to, is of critical importance. It is important to note, that the nodes in this protocol are free to choose from the array of available tip selection methods. In this paper we focus on two tip selection algorithms:

Firstly, the *Uniform Random Tip Selection* (**URTS**), is a very basic algorithm: we simply select a tip from the set of all available tips with a uniform random distribution. Despite being the most efficient method numerically, it would also be accompanied by security vulnerabilities and allow for tip selection behavior that is non-beneficial for the safety of the Tangle [27]. However, as we will see in Section 2.2 it is closely related to the next algorithm, which is the one that most resembles the current implementation in the protocol. More particularly, the analytical derivations for the following algorithm depend on the solutions for this URTS algorithm.

The second tip selection algorithm employs a Monte Carlo Markov Chain: here we select the tip at the end of a random walk (**RW**), beginning at the first tx in the Tangle. In the current implementation of the IOTA protocol, the RW can be biased towards transactions with large cumulative weight, which is a tx's own weight plus the sum of all own weights of directly or indirectly approving txs. The amount of bias is determined by a parameter α. When $\alpha = 0$, we dub the RW as *Unbiased Random Walk* (**URW**). Consequently, when $\alpha > 0$, we dub it *Biased Random Walk* (**BRW**). As we will see in Section 2.3 under the current network conditions and implementation of the IOTA protocol, the BRW has very similar properties to the URW and, therefore, it is sufficient to study the URW.

1.2   The Parasite Chain attack

Due to the probabilistic nature of the immutability of the ledger state, cryptocurrencies are subject to certain security concerns [21]. One such concern is that an adversary may revert the ledger to an earlier state if he possesses a sufficient amount of hashing or voting power [28]. In practice, such an adversarial action would result in a fork of the ledger and enable the possibility for a double-spend of funds.

[27] describes several attack scenarios under which an adversary may attempt a double-spend on the IOTA protocol. In this paper we focus on the Parasite Chain (**PC**) attack. In this attack the adversary places a value tx in the main Tangle, whilst also creating a

a) Tangle with no Parasite Chain

b) Tangle with Parasite Chain revealed

Double Spend
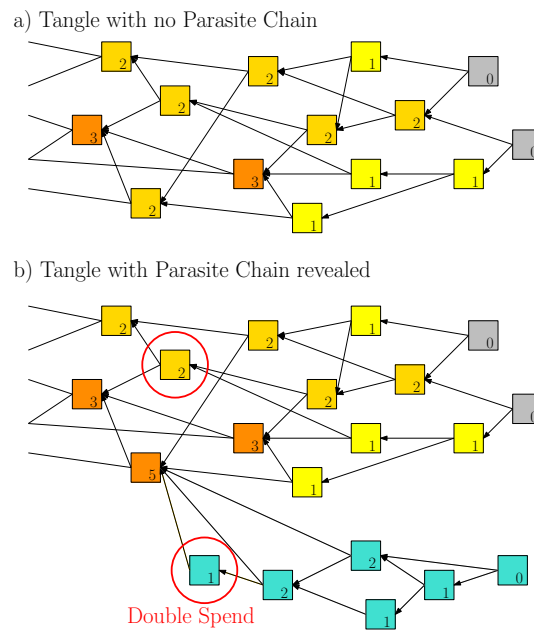
Figure 1 Examples of parts of a Tangle. Transactions are represented by squares and approvals by arrows. The number of approvers of each transaction is in the bottom right corner of each transaction. Parts (a) and (b) represent the Tangle with and without a conflicting Parasite Chain attached. Transactions that conflict with the most recent part of the main Tangle are shown in blue.

side chain in secret that contains a double-spending tx, see Fig. 1. Once the PC is revealed to the network, the attacker then exploits the tip selection algorithm by leading honest txs to approve the PC instead of the main Tangle. This is possible, through the following mechanism: firstly, by attaching the PC to a particular tx (the *root* tx), the cumulative weight of this root tx can be significantly increased and the RW tip selection algorithm will then be drawn towards this root. In addition, the attacker can also increase the number of links from the PC to this root tx, to increase the probability for the RW to continue onto the PC. If the attack is successful, the part of the Tangle that approves the originally visible double-spend tx is abandoned and the PC becomes the new main Tangle, thereby changing the ledger history.

[11] discuss this attack in further detail and show that for certain values of the parameter α in the BRW, the attack has an increased likelihood to succeed. Furthermore, although the security is improved with increasing value of α, it should not be selected too high, otherwise txs would be left behind and excluded from the ledger. Therefore, α has to be selected in a manner that is a compromise between the safety of the system and avoiding orphanage of txs, i.e. txs not being approved. More generally [11] also showed that the success of the attack also depends on other variables, such as the time that the PC is revealed to the main tangle and the number of root txs the PC is attached to the main Tangle.

In this paper, we present a method to counter the vulnerability of the PC attack by introducing certain detection mechanisms. These detection methods employ the knowledge of the underlying structure of the Tangle, more particularly the likelihood for txs to have a certain number of direct approvers. We show that by measuring the distribution of approvers for a selected set and comparing it to a reference distribution, the Tangle can be checked for abnormalities. Upon flagging a suspicious part of the Tangle as a PC, countermeasures can be taken, such as restarting the RW with an increased α value [25]. Since the tip selection

method can be switched or restarted immediately during the detection, an honest tx issuer who detected the PC, would avoid approving tips of the PC and hence contribute towards rendering the attack unsuccessful. We show that an adversary would have to significantly reduce the efficacy of the attack if he wants for the PC to remain undetected.

1.3   Contributions of this paper

Our contributions in this paper are twofold: firstly, we present analytical models that capture the underlying structure of the Tangle, in the form of the likelihood for txs to have a certain number of direct approvers. We present models for the URTS and the URW, and we show that the distribution for the URW is tightly linked with the distribution for URTS. We compare the analytical models to simulation results and show that generally a good agreement is reached in the high load regime. In the low load regimes, the model predicts the values well only if txs are only allowed to approve the same tx once.

Secondly, we describe a method of how to reduce the susceptibility of the IOTA cryptocurrency towards a specific type of double-spend attack, more particularly a Parasite Chain. This enables a proactive tool against malicious actors and improvements for security. We measure how 'distant' a certain sample set of txs is from the derived distributions and we show that we can employ the distance metric, to effectively detect simple versions of a PC. If the attacker decides to avoid the detection methods, he is forced to build the PC in a more complicated way, which is correlated to a decrease of the attack's efficacy. We demonstrate this for two ways of selecting sample sets: through RWs and through collecting the txs that directly or indirectly approve a particular tx, i.e. the future cone of that tx. We conclude that by combining these two methods, a powerful detection tool is provided to honest tx issuers that allows them to make their tip selection more safe and PC attacks less likely to succeed.

2   Model for the Number of Approvers

Due to the complexity of the Tangle, it can be difficult to derive exact solutions which describe certain mechanics. On the other hand, deterministic solutions can provide a sufficiently good picture to describe certain mechanics despite their simplified assumptions. Here, we attempt to discuss the likelihood of a randomly selected tx having a given number of direct approvers n, through a deterministic model. The model is facilitated by employing a linear approximation of the exit probability distribution, which represents the likelihood of a tip being selected.

Since the tip selection is performed probabilistically, the same tip may get selected twice, although this is only likely to happen in the *low load regime*, i.e. if the rate of arriving txs is low. Since the tip selection algorithm is not enforced, it remains up to the node to decide what to do in this case. Here, we consider two particular scenarios that do not require rerunning the tip selection algorithm and which are, therefore, considered numerically efficient options. Initially, we will discuss both scenarios, before focusing on the former in more detail: in the single edge model (**SEM**) only one edge is created instead of two, and in the multi-edge model (**MEM**) two edges are created to the same tx. As we will show, in the *high load regime*, i.e. when the rate of arriving txs is high, SEM and MEM converge to the same distributions. Since the Tangle is built to allow for high throughput, we will mainly focus on the high load regime and the difference between SEM and MEM can be neglected. However, since the protocol should also be analyzed in the low load scenario and the node is free to choose the tip selection algorithm, the two methods are discussed for completeness and comparison.

Once a tx is selected for the tip selection, it is assumed that the Proof-of-Work plus the propagation to the network equal to the time h, and that no other node will be aware of the approval before this delay has passed [27]. Let p_i be the probability of being selected by the tip selection algorithm for the i-th tx, N_i the final number of approvers of i and t_i the time of first approval. We notice that p_i depends on the number of tips and hence may change with the arrival of a new tx. Note that if txs i_1 and i_2 are the approvees of i then no further txs are attached to them later than $t_i + h$, due to the delay h. It is noteworthy that the following applies for URW: after $t_i + h$ none of the directly or indirectly approved txs, i.e. the entire past cone of i, receive any further approvers. The exit probability of the URW at i remains then unaffected after $t_i + h$ by the arrival of new tips and p_i, and also the probability for the random walk to pass through the tx, remains constant after that time. This is not the case if backtracking is allowed, i.e. the RW is allowed to return to the past cone of i once it left it, however, this is currently not implemented in the IOTA protocol.

Throughout this work we will frequently employ the Poisson distribution function

$$P(\gamma, n) = e^{-\gamma} \frac{\gamma^n}{n!} \tag{1}$$

where γ is a rate.

In the following sections we employ

▶ **Lemma 1.** *The number of approvers for a given tx i is given by*

$$N_i = 1 + p_{i0} + Pois(\lambda_i) \tag{2}$$

where $\mathrm{Pois}(\cdot)$ *is the Poisson distribution,*

$$\lambda_i = \lambda \int_{t_i}^{t_i+h} dt \begin{cases} 2p_i(t) & \text{for MEM} \\ 2p_i(t) - p_i(t)^2 & \text{for SEM} \end{cases} \tag{3}$$

λ *is the tx rate in units of h, and*

$$p_{i0} = \begin{cases} p_i(t_i) & \text{for MEM, approximately} \\ 0 & \text{for SEM} \end{cases} \tag{4}$$

Proof. Assume there is a Poisson process of arriving txs [27] with rate λ, i.e. the number of arrivals within the interval h after i receives its first approval is given by the random variable N. Let us consider SEM first. The probability of i receiving an additional approver, once a tx arrives is $p_i^+(t) = 1 - (1 - p_i(t))^2$. Hence from the viewpoint of i, txs arrive at rate $p_i^+\lambda$ (of which all would reference i). Furthermore, for independent events it holds that $\mathbb{P}(N = n) = \mathbb{P}(M_1 + M_2 = n)$, where $\mathbb{P}(X = x)$ is the probability that the random variable X takes the value x, and M_1 and M_2 are again Poisson processes, with rates μ_1 and μ_2. We can, therefore, assume that the arrivals of attachments to i occur through a series of Poisson processes with rate $p_i^+\lambda dt$ at time intervals dt, which leads to the integral form. In the case of MEM, we consider two rates of arriving tx that approve i: txs that approve i twice or once, and their rates of arrival are $p_i(t)^2\lambda$ and $(p_i^+(t) - p_i(t)^2)\lambda$, respectively. With the same argument as above, we can find the integral form

$$N_i = 1 + \mathrm{Pois}(\lambda_{i1}) + 2\mathrm{Pois}(\lambda_{i2})$$

where

$$\lambda_{i1} = \lambda \int_{t_i}^{t_i+h} 2p_i(t)(1 - p_i(t))dt$$

$$\lambda_{i2} = \lambda \int_{t_i}^{t_i+h} p_i^2(t)dt$$

These two Poisson processes are independent and can be further combined. Finally, the first approver also has the likelihood p_{i0} to approve the same tip twice. This assumption does not take into account that for small λ the Poisson process has a significant impact on the number of approvers. More specifically, for small λ the likelihood is increased that a single tip is simultaneously selected by one or multiple txs and hence the probability to have an even number of approving edges is increased compared to an odd number.   ◀

Generally the average probability to have n approvers is given by

$$P(n) = \sum_{i=1}^{N} \mathbb{P}(N_i = n) \tag{5}$$

where N is the cardinality of a list of txs that are considered. Since as previously discussed for URW $p_i(t)$ is only variable for a short time (maximally h after i is revealed), we assume $p_i(t) = p_i \forall t$ is approximately true. The integral forms in (3) can, therefore, be further simplified, which we employ in the next section.

2.1   Uniform Random Tip Selection

In the URTS algorithm, tips are selected at random from the set of L available tips and with the most recent assumption $p_i = L^{-1}$ on average. According to [27], the number of tips is approximately $L = 2\lambda$. However, for small λ, the number of tips is limited by one instead. For simplicity, we assume that

$$L = 1 + 2\lambda \tag{6}$$

Using (5) and (2) the probability to have n approvers (or equivalently $n - 1$ beyond the first) is then given by the Poisson distribution function

$$P_U(n) = P(\lambda_U, n - 1) \tag{7}$$

with the rate

$$\lambda_U = \begin{cases} 2\lambda L^{-1} & \text{for MEM} \\ 2\lambda L^{-1}(1 - 0.5L^{-1}) & \text{for SEM} \end{cases}$$

Note that in the high load regime (i.e. λ is large), the quadratic term can be neglected and MEM and SEM converge to the same rate, as previously discussed.

Fig. 2 shows a comparison of the numerical and analytical values of $P_U(n)$ for both SEM and MEM. It can be seen that for SEM the model represents the numerical values well. In both models, the distribution converges towards the same distribution in the high-load regime. This is because for large λ almost none of the tips are selected twice. However, in the low load regime (i.e. $\lambda < 10$) the predicted values do not match for MEM. This is due to the discrete nature of the Poisson process, which as described above, is not accounted for. Furthermore, even numbers of approvers have a higher probability (see $n = 4$), while odd numbers of approvers are less likely than predicted (see $n = 3$).

2.2   Unbiased Random Walk

The tip selection algorithm that is currently employed in IOTA is the Monte-Carlo-Markov-Chain RW with the parameter α [27]. Typically, analytical solutions are easier to find by initially considering $\alpha = 0$, where effects such as txs that are left behind, or the dependence
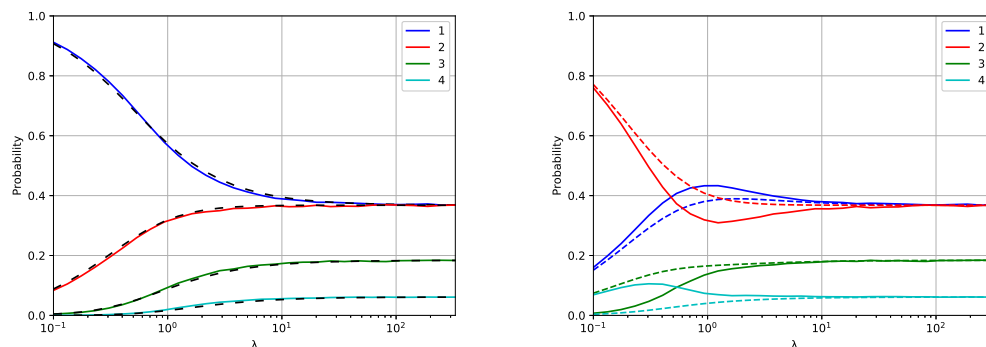
a) Single-Edge model (SEM)   b) Multi-Edge model (MEM)

Figure 2 Probability for a randomly selected tx, in a Tangle created by URTS, to have n approvers with λ for $n = \{1, .., 4\}$. Values from simulation results and analytically predicted values are shown with continuous and dashed lines, respectively.

of the RW on the cumulative weight, play no role. Here we follow the same path, before increasing the complexity of the analysis by considering $\alpha > 0$ in Section 2.3. We employ SEM for the analysis in this section, since the analytical model provides more accurate results in this case. However, for most of our analysis, both models would be appropriate, since both predict the numerical values well in the high-load regime, which is in the focus in this work. Note that [3] observed that for $\alpha = 0$, the mean tip number converges to $L \approx 2.1 \cdot \lambda$. However, employing this observation does not lead to improvements for this model and hence the same average tip number as for URTS is employed.

For a given number of tips L, we can order the exit probabilities by their likelihood. We then define the L-normalized exit probability $e(x)$ to be the exit probability distribution that is normalized from the index interval $\{1, .., L\}$ onto the relative index interval $(0, 1]$. For large enough λ, the expected exit probability of the $i - th$ most likely tip is then approximately given by

$$\mathbb{E}e_L(i) \approx \int_{(i-1)/L}^{i/L} e(x)dx \tag{8}$$

The L-normalized exit probability can be expressed as

$$e(x) = 1 + f(x) \tag{9}$$

where $x \in [0, 1]$, $\int_0^1 f(x) = 0$ and $f(x) > -1$.

Fig. 3 shows the numerically calculated L-normalized exit probability for URTS and URW, as well as a linear fit for URW. $e(x)$ has numerically been calculated by averaging over 10^3 samples of tip sets. For each tip set, 10^6 tip selections are performed and the tips are ordered by their exit probability $e_L(i)$. Despite the high number of tip selection samples, the values of indices close to 0 (1) are slightly over- (under-) estimated due to stochastic effects, as can be seen for URTS where the value is expected to be $e_U(x) = 1$ over the entire interval. It can be seen that for the fitted curve the numerical values agree well for most of the range, apart from the smaller indices, where the exit probability is noticeably increased, suggesting that certain tips in the Tangle have a considerably higher probability to be selected.
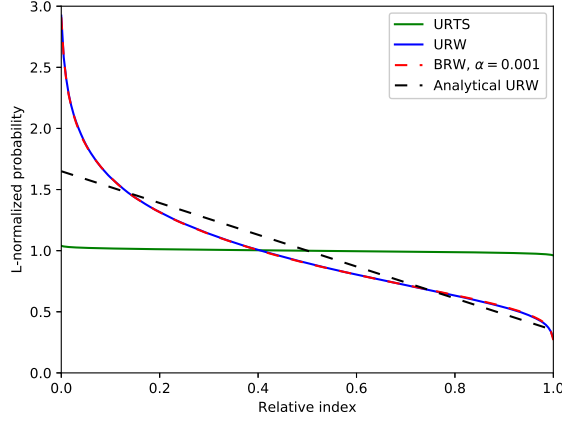
Figure 3 L-normalized exit probability with the ordered relative index number for $\lambda = 100$. The curve for the analytical linear approach is shown for $a = 1.3$.

For a given relative index number x, we can assume a Poisson process of txs and the probability density of having n approvers is then given by

$$p_{URW}(n, x) = P(e_{URW}(x)\lambda_U, n - 1) \tag{10}$$

The expected value is then given by

$$
\begin{aligned}
P_{URW}(n) &= \int_0^1 dx\ e^{-e_{URW}(x)\lambda_U} \frac{(e_{URW}(x)\lambda_U)^{n-1}}{(n-1)!} \\
&= P_U(n) \int_0^1 dx\ e^{-f(x)\lambda_U} (1 + f(x))^{n-1}
\end{aligned} \tag{11}
$$

As can be seen, the distribution for the URW can be decomposed into a product of the distribution for URTS times a factor dependent on the exit probability function. This highlights how these two tip selection algorithms are correlated, and why their distribution appears similar.
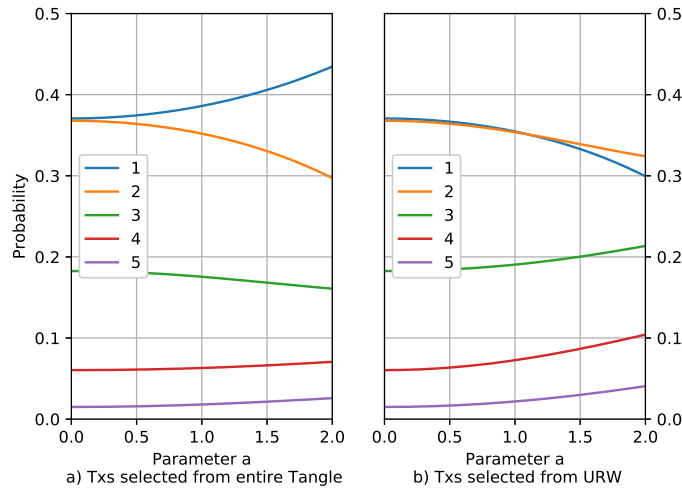
For URW, the probability to pass through a given tx, is equal to the final value of the exit probability during the time the tx was a tip. Hence to derive the approver statistics observed along the path of RWs, (10) is weight by $e(x)$, and the expected distribution is

$$P^*_{URW}(n) = \frac{1}{b} \int_0^1 dx\ p_{URW}(n, x)e(x) \tag{12}$$

where

$$b = \sum_{n=0}^{\infty} \int_0^1 dx\ p_{URW}(n, x)e(x)$$

normalizes the probability.

Figure 4 Values of (13) and (14) with the parameter a for several values of approver numbers n. $\lambda = 100$.

Linear approach

A simplified approach is presented by employing the distribution $f(x) = a(x - 0.5)$, $a \in [0, 2]$, where a is limited to achieve only positive values for f. The expected value is then given by

$$P_{URW}(n) = P_U(n)g(n - 1) \tag{13}$$

$$P^*_{URW}(n) = \frac{P_U(n)g(n)}{\sum_{m=1}^{\infty} P_U(m)g(m)} \tag{14}$$

where

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

Note that for $a \to 0$ (13) converges towards URTS:

$$\frac{P_{URW}(n)}{P_U(n)} = -\sum_{j=0}^{n-1} \frac{(n-1)!}{(n-1-j)!}(n-2-j) = 1$$

Fig. 4 shows the probability to have n approvers with the value of the parameter a. As can be seen in Fig. 4a), introducing a gradient in the exit probability shifts the likelihood of having a certain number of approvers away from the mean towards less (one approver) or higher numbers of approvers (more than 3), corresponding to increased or decreased exit probabilities. On the other hand, it can be seen from Fig. 4b) that the likelihood for a URW to visit sites with a higher number of approvers, is shifted to only txs with higher numbers (more than 2), since these are sites more frequently visited by the URW.

From Figs. 3 and 5, we can see that a good agreement is achieved with the numerical results for $a = 1.3$.
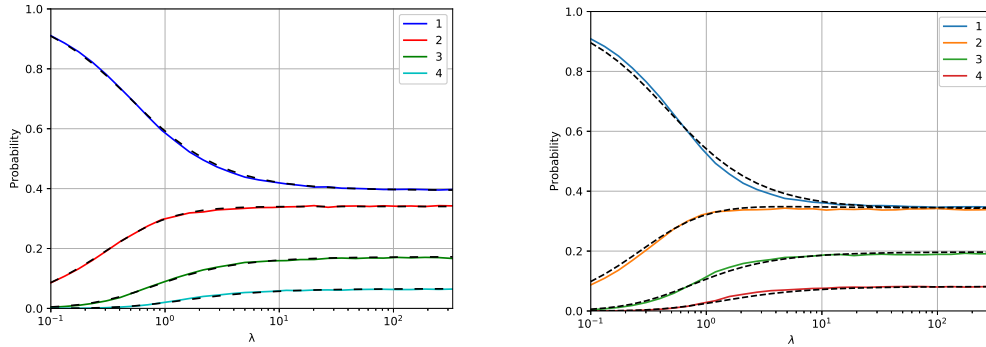
a) Randomly selecting a tx from the entire Tangle

b) Randomly selecting a tx from an URW

■ **Figure 5** Probability for a tx to have n approvers with λ for $n = \{1, .., 4\}$; for SEM and URW tip selection, and $a = 1.3$. Values from simulation results and analytically predicted values are shown with continuous and dashed lines, respectively.

2.3   Biased Random Walk

In the IOTA protocol, the RW is made dependent on the cumulative weight of a tx. The cumulative weight w_x of a tx x is defined as the sum of its own weight, plus all own weights of txs directly or indirectly approving the tx. The dependency on the RW is introduced to prevent undesirable behaviors, such as the selection of old tips (lazy tip selection), or certain types of PCs [27]. In the current implementation of the Tangle, the path of the RW depends, therefore, on a parameter α and the cumulative weight of txs encountered along the path. More specifically, the probability to transition from a tx x to a tx y is given by [27]:

$$P(x \rightsquigarrow y) = \frac{\exp(\alpha w_y)}{\sum_z \exp(\alpha w_z)} \tag{15}$$

where the denominator normalizes the probability by summing over all direct approvers z (including y) of x.

It should be noted that the value of the parameter α ought to be selected carefully. Selecting the value too low would result in the BRW having little or no improvement to the security compared to URW, while a too high value would result in txs being orphaned. Typically, orphanage behavior is observed in simulations at about $\alpha\lambda > 1$ [18]. The default value in the IOTA reference implementation of $\alpha = 0.001$ [1] ensures that the Tangle structure is the same at least up to a tx rate of $\lambda = 100$, since no orphans are created and the exit probability is the same, as can be seen in Fig. 3.

With the Minimum Weight Magnitude parameter set to 14 (current setting of the difficulty in the IOTA protocol), the average time for the PoW in a Core i7 platform is about $4.1s$ [2]. At a tx rate in the order of 5 tps (current, over the scope of one day, measured tps by an IRI node [1]), the expected value for the tx rate is $\lambda \approx 20$. Since $\lambda \gg 1$, the difference between SEM and MEM is negligible. Furthermore, since $\lambda\alpha \ll 1$, the difference between the exit probabilities and hence the approver distributions of URW and BRW are negligible, as discussed in the previous paragraph. We, therefore, employ the models developed in Section 2.2 for the following sections.

3 Parasite Chain detection

The vulnerability of the Tangle to the PC attack is first recognized in [27]. A PC is a specific type of a double-spend attack, where the attacker issues a tx on the main Tangle and, simultaneously, issues a double-spending tx, which is not yet revealed. He then continues to issue further txs that approve the double-spend on the PC in secret, to strengthen the PC chain, till the receiver accepts the main Tangle double-spend tx. Upon receiving the goods or value from the to-be-swindled receiver, the adversary then reveals the PC in the hope that it can outpace the incompatible part of the current main Tangle, by attracting as many tip selection RWs as possible. If successful, the part of the Tangle that approves the main Tangle double-spend tx becomes orphaned, and the PC becomes the new main Tangle, thereby invalidating the main Tangle double-spend tx.

In this paper, we focus on a specific Parasite Chain attack where the adversary pins the PC to one or more (k) honest txs in the main Tangle, which are issued at a similar time as the double-spend tx on the main Tangle. We dub this type of a PC k-pinned PC. Generally, an adversary may choose to attach to multiple root txs. However, for simplicity we assume a 1-pinned PC.

We assume that the adversary attempts to issue as many txs as possible that directly approve the PC root, as well as the previously issued PC tx. This approach is taken for two reasons: firstly, by directly or indirectly approving this root tx, the cumulative weight is increased both by the txs in the main Tangle, as well as by the PC. This can significantly increase the cumulative weight of the root, compared to the double-spend tx in the main Tangle. As a result, the RWs that are performed for tip selection will be drawn towards the root. Secondly, by attaching as many direct approvers to the root, the probability for the RW to hop onto the PC once the RW passes through the root, depends on the number of links to the PC. It can, therefore, be increased by directly attaching as many malicious txs as possible. Fig. 6a) shows the most efficient way of attaching a 1-pinned PC, and we dub it a simple PC (**SPC**).

In this section, we describe a methodology for how we can employ the distributions that we derived in the previous section, to detect a pinned PC. More particularly, since the distribution of approvers in the PC in Fig. 6a) is significantly different from the distributions derived in the Section 2, we propose the implementation of a metric that compares the encountered approver distribution to the expected one, and allows to identify PCs. This would allow nodes to actively counter the creation and success of PCs.
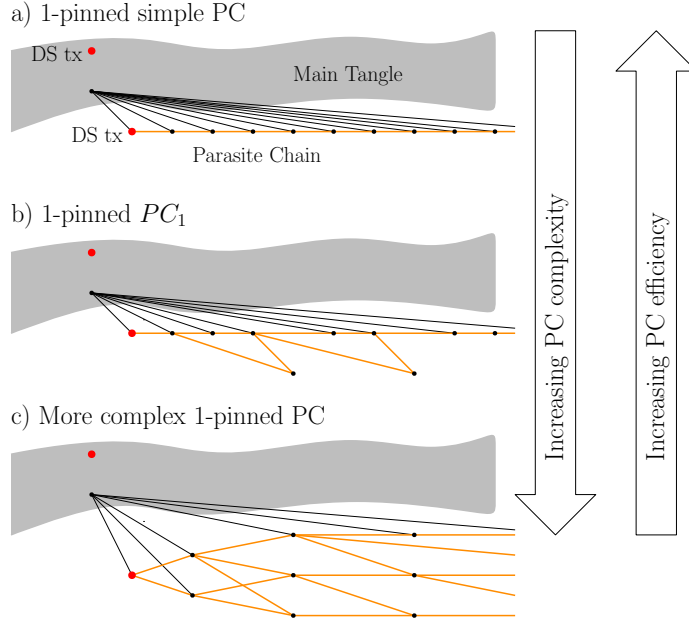
We define the following distance metric

$$d_P = \frac{1}{2} \sum_{n=0}^{\infty} |P(n, S) - P_{ref}(n)| \tag{16}$$

where P is the measured distribution, or probability vector, that is deduced by measuring on the sample of size S. In this paper, the sample size is varied between 10 and 100, see Fig. 7. P_{ref} is the reference distribution (equations (13) or (14)) and the factor $\frac{1}{2}$ normalizes the distance.

As discussed, it is more expensive for the attacker to create txs in the PC with higher numbers of approvers. Hence, it is useful to put increased weight on txs with higher numbers of approvers. However, due to the low probability of having a high number of approvers, the difference between the actual value $P(n)$ and $P_{ref}(n)$ can be relatively large for higher values of n. To reward having a high number of approvers (instead of penalizing it because of the high variance), we measure if a tx has more than a given number of approvers, and we

a) 1-pinned simple PC

DS tx

Main Tangle

DS tx

Parasite Chain

b) 1-pinned PC_1

c) More complex 1-pinned PC

Increasing PC complexity

Increasing PC efficiency

Figure 6 Different types of PCs with an increasing effort of the adversary to hide the PC from the detection methods. The number of indicated malicious txs is kept the same for a)-c). With higher complexity, fewer links to the root tx in the main Tangle are created.

can therefore also employ the distance

$$d_Q = \frac{1}{2} \sum_{n=0}^{\infty} |Q(n) - Q_{ref}(n)| \qquad (17)$$

where

$$Q(n) = \sum_{m=n}^{\infty} P(m) \qquad (18)$$

is the cumulative probability distribution and Q_{ref} is the reference cumulative probability distribution. As will be shown in the following figures, employing d_Q instead of d_P can lead to a better analysis, whilst adding little or no computational overhead for calculating the more complicated metric d_Q.

We can employ these metrics in the following way: if the measured distance exceeds a critical value η, the detection method rejects this path for the RW. More formally, we say a sample P fails an η-confidence level test if $d_P > \eta$ (or equivalently, the sample Q fails if $d_Q > \eta$). The exact procedure after rejecting a path can be manifold. For example, the node may simply restart the RW from the last point where no suspicious behavior was noted. Or the node may also switch into some kind of safe mode, where the RW is calculated with a higher α value [25].

Generally, the measurement of the distance is made on a limited sample size and the distances d_P and d_Q can, therefore, only take certain discrete values. To determine how likely it is that a measurement is treated as a false-positive, i.e. the sample is part of the main Tangle but has been flagged to be in a PC, we study how likely it is that a certain distance occurs in a random sample. Due to the discreteness of the above distance metrics, we investigate the cumulative probability rather than the actual distribution. Note that for small

■ **Algorithm 1** Integration of detection into tip selection.

---

    input   **:** The starting point of the RW defined as initial_tx
    output : The selected tip

1 tx ← initial_tx
2 mode ← standard
3 **while** *tx is **NOT** a tip* **do**
4   **if** *mode = safe* **then**
5   tx ← safe_step(tx)
6   **else**
7   tx ← standard_step(tx)
8   **if** *Algorithm 2 = TRUE* **then**
9   mode ← safe
10   tx ← initial_tx

11 **return** tx

---

sample sizes S, this discreteness can be observed through the clearly visible discontinuity of the derivative of the curves, see e.g. Fig. 7 for $S = 10$. Intuitively, the cumulative probability also provides a visual representation as to how many samples would be below a given distance, and how many false-positives we should expect, when setting η.

3.1 Random walk detection

In this method of creating a sample of approver numbers, we collect and update the sample set of txs while the RW traverses the Tangle. This method is numerically very inexpensive, since the information for the sample set is readily available, as it is already considered when performing the RW. Hence the node may also decide to test samples of different lengths in parallel, in order to distinguish between very local distributions (small sample size, e.g. $S = 10$) and more global distributions (e.g. $S > 50$).

The pseudo code described in Algorithm 1 provides an example high-level overview of how the PC detection mechanism could be implemented in a tip selection algorithm. The additional algorithm 2 is added, which updates and evaluates for every RW step a sample of approver numbers. If the calculated distance d is larger than a given threshold, the algorithm 2 returns a flag and the tip selection method restarts, in this case, from the initial starting point of the RW. Furthermore, the RW is changed from the standard RW (with a *standard_step*) into a safe mode (*safe_step*). This safe mode could, for example, be a RW with an increased α value, as described in [25].
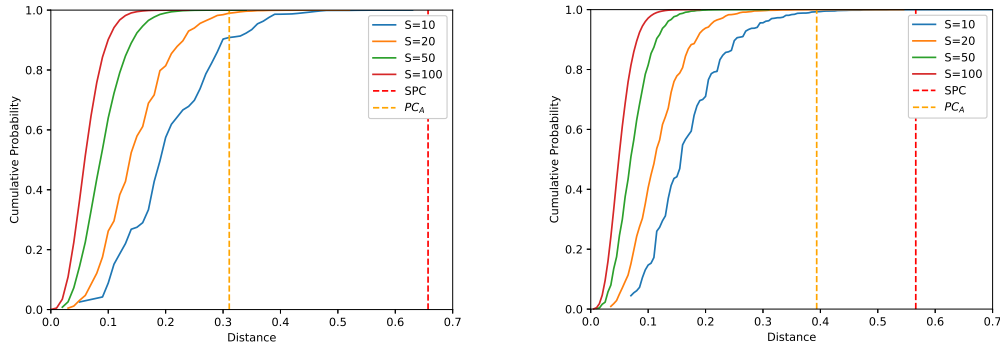
Fig. 7 shows the cumulative probability for the distance to be above a certain value, where the sample size of considered txs has been fixed to S and P_{ref} is given by (14). It can be seen that for small values of S the discrete nature of the distance is clearly visible, while for larger S it becomes increasingly difficult to observe this phenomenon. Note, that the maximum available number of samples from an RW depends on the depth from which the RW is started. Furthermore, S should be selected such that local changes (i.e. the start of a PC) can be detected, which puts further restrictions on the upper limit of S.

Let us assume the adversary attaches a 1-pinned PC to the Tangle, and that with probability p_R a malicious tx directly approves the PC root, as well as the previous malicious tx that was attached in the same manner. We denote the part of the PC that is constructed

Algorithm 2 Sample Management.

---

input : current tx, sample list of approver numbers
output: bool value wheter PC is detected

1 Add number of approvers of tx to the sample list
2 if *Sample size > S* **then**
3 Remove the oldest element from the sample list

4 Calculate distance d by applying Eq. (16) or (17)
5 pc_detected ← (d > threshold)
6 return pc_detected

---

a) Distance metric d_P

b) Distance metric d_Q

Figure 7 Cumulative probability for the measured distance to be below a given value for several sample sizes S. $\lambda = 100$, URW.

in this way as the *main PC*. The remaining malicious txs are attached in whatever way seems most suitable. Then, the most efficient way the adversary can build the PC is with $p_R = 1$, see Fig. 6a), where all malicious txs are added as direct approvers to the PC root and the rate at which direct approvers are added to it is given by $r = \mu$.

As can be seen from Fig. 7, an SPC is easily detectable, since even for a small number of samples S, the measured distance in the PC is larger than the measurements in the main Tangle. By decreasing p_R and attaching to txs along the main PC (1-pinned PC_1, see Fig. 6b)), the adversary may render the PC undetectable. However, by doing so he weakens the attack since fewer edges are attached to the PC root. Note also that in the PC_1 in Fig. 6b), txs that are not part of the main PC are left behind and due to their low cumulative weight, are therefore unlikely to be selected by the RW as tips.

The selected sample size S should be sufficiently large, since otherwise the detection method may show little success (or return too many false-positives, if the value of d_C is set too low). For example, denote the following PC as PC_A: an adversary employs the strategy PC_1, by adding a link to the main PC with the probability $1 - 0.5P^*_{URW}(2)$. Then

$$d_P = 1 - P^*(1) - P^*(2) \approx 0.26$$

and the attachment to the root is reduced to the rate

$$r = (1 + 0.5P^*_{URW}(2))^{-1}\mu \approx 0.85\mu$$

If S is selected too low (e.g. $S = 10$) this type of PC may remain undetected, if d_P (equation

16) is employed for detection, as can be seen from Fig. 7a). However, it still may be detectable when changing the detection metric to d_Q (equation 17).

The PC would certainly remain undetected, if txs are added to the main PC, such that the encountered distribution is $P^*_{URW}(n)$, i.e. $d = 0$. Under this condition, the rate is reduced to

$$r = (1 + \sum_{n=1}(n - 1)P^*_{URW}(n))^{-1}\mu \tag{19}$$

For example, for $\lambda = 100$ and $\alpha = 0.0001$, $r \approx 0.46\mu$, which requires the attacker to deploy the malicious txs in a much less efficient way, hence reducing the efficacy of the entire PC attack.

3.2   Future cone detection

In a pinned PC, the capability of those txs that are pinned to the root, to divert the RW onto the PC relies on their cumulative weight growing as fast as possible. In other words, it is likely that most of the issued malicious txs reference, directly or indirectly, earlier malicious txs. An effective method to consider most of the PC txs is, therefore, to analyze the future cone of the txs that are encountered along an RW, where the future cone of a tx is defined as the set of txs that directly or indirectly approve a given tx.

Note that compared to the method in the previous section, the sample size S can be significantly larger, since the number of samples in the main Tangle would increase exponentially with the distance from the cone's root tx. Until the growth of the future cone reaches the linear phase, where the rate of txs per unit of distance to the root would be constant. However, while the method in Section 3.1 is computationally relatively inexpensive, here we must employ a traverse algorithm to efficiently collect a sample of txs, such as Breath-First Search or Depth-First Search [13]. It is, therefore, recommended to employ this control mechanism only occasionally or if suspicion is raised.

Due to the larger possible sample size and the way it is sampled, the adversary must attempt to achieve a better agreement of the PC with the reference distribution (13). For example, the relatively inexpensive PC in 6b) would exhibit a high distance d_P (d_Q), due to the suspiciously large amount of orphans.

The adversary may still create a PC structure that makes the PC less likely to be detected, as Fig. 6c) indicates. However, since the future cone envelopes the distribution of all txs within a certain distance of the investigated root tx (and in the future cone), the necessary structure becomes significantly more complicated. More specifically, since the reference distribution itself stems from a structure where all links are employed, this leaves effectively little to no spare links for the adversary to attach to the PC root: due to the fact that the PC is created in secret, the only approvers that the txs can receive before being revealed are the txs from within the PC itself. Hence the average number of approvers in the PC is

$$\bar{n}_{PC} = \sum_n nP_{PC}(n) \leq \sum_n nP_{ref}(n) = 2$$

where P_{PC} is the probability distribution within the PC and the inequality arises, since the PC creates links to txs within the main Tangle, without receiving any in return. The second equality in the above equation is due to the fact that the total average number of edges per tx in the Tangle must be equal to the number of approvers a tx selects, which is two in the high load regime. Hence, the PC structure that provides optimal resistance to the detection method would require most links to be employed within the PC. The adversary must, therefore, choose a medium between detectability and efficacy of the PC, by deploying links purely within the PC, i.e. he cannot create direct links to the PC root.

In conclusion, the main observation is that, contrary to the approach in Section 3.1 where adding a few tx according to the method in 6b) may be sufficient, in this approach it is more difficult to reproduce the reference distribution. More precisely, in order to imitate the exact distribution, all adversary txs would need to be deployed for mimicking the distribution.

3.3 Final remarks

Further improvements can be envisioned for the presented detection tools. For example, we may combine the RW detection in Section 3.1 and the future cone detection 3.2, since their distributions are different as can be seen by comparing Figs. 5a) and 5b). Avoiding both detection mechanisms may be noticeably more difficult and their combination could, therefore, lead to an improved success rate. The numerically expensive method in Section 3.2 may also be more suitable for taking larger samples, while the method in Section 3.1 can be easily implemented without much additional numerical effort. Furthermore, the sample size could be continuously varied and alternative sampling mechanisms could be envisioned, which would make it more difficult for the adversary to adopt his strategy.

4 Conclusion

We present models to understand and predict a particular aspect of the Tangle structure, more specifically, the likelihood for a transaction to obtain a particular number of direct approvers. We derive solutions for two tip selection mechanisms: firstly, for a Tangle that is built employing a uniform random selection and secondly, if it is constructed employing a random walk. We show that the distributions for the two tip selection methods are linked tightly and also that the distribution depends on whether samples for transactions are taken from the set of all transactions, or only from transactions which are encountered along random walks.

We then employ the models for the approver distributions to detect a specific double-spend attack on the IOTA cryptocurrency, namely the *Parasite Chain* attack. In this type of attack, an adversary attempts to trick a node's tip selection mechanism into approving a double-spend transaction, by building a side Tangle in secret and revealing it at a suitable moment. Since the efficacy of the attack relies on building a side tangle that directly approves a limited set of transactions in the main Tangle, the underlying structure of that Parasite Chain is noticeably different to the main Tangle. By measuring the *distance* to the derived distributions on direct approvers, it is possible to detect certain forms of this side chain. It is shown that the quality of the detection depends on the sample size and we, therefore, propose two different methods of sampling. Firstly, a numerically inexpensive method is proposed, where the node may record the approver statistics along the path of a random walk. However, since the random walk moves relatively fast through the Tangle, a limited amount of transactions can be sampled. In the second approach, the node chooses to sample the future cone of a given transaction. This would ensure that most of the transactions in the Parasite Chain can be measured against the expected distribution.

Through these methods, certain structures of the Parasite Chain may be detectable. This would allow the honest tx issuers to improve their tip selection algorithm to be more safe and may prevent Parasite Chain attacks from being successful. On the other hand, if the adversary chooses to avoid detection by constructing more complex forms that would be more difficult to detect, he has to deploy a significant proportion of transactions in a less effective manner. In either case, the proposed methods present a tool which can reduce the threat imposed by Parasite Chains.

───── **References** ─────

1   https://github.com/iotaledger/iota.go.

2   Elsts A., Mitskas E., and Oikonomou G. Distributed Ledger Technology and the Internet of Things: A Feasibility Study, 2019.

3   Kuśmierz B., Staupe P., and Gal A. Extracting Tangle Properties in Continuous Time via Large-Scale Simulations, 2017.

4   Iddo Bentov, Pavel Hubáček, Tal Moran, and Asaf Nadler. Tortoise and hares consensus: the meshcash framework for incentive-compatible, scalable cryptocurrencies. Cryptology ePrint Archive, Report 2017/300, 2017. URL: `https://eprint.iacr.org/2017/300`.

5   Xavier Boyen, Christopher Carr, and Thomas Haines. Blockchain-Free Cryptocurrencies: A Framework for Truly Decentralised Fast Transactions. Cryptology ePrint Archive, Report 2016/871, 2016. URL: `https://eprint.iacr.org/2016/871`.

6   Vitalik Buterin et al. A next-generation smart contract and decentralized application platform, 2014. URL: `https://github.com/ethereum/wiki/wiki/White-Paper`.

7   Vitalik Buterin and Virgil Griffith. Casper the Friendly Finality Gadget. *ArXiv e-prints*, page arXiv:1710.09437, October 2017. `arXiv:1710.09437`.

8   Tai-Yuan Chen, Wei-Ning Huang, Po-Chun Kuo, Hao Chung, and Tzu-Wei Chao. DEXON: A Highly Scalable, Decentralized DAG-Based Consensus Algorithm. *ArXiv e-prints*, page arXiv:1811.07525, November 2018. `arXiv:1811.07525`.

9   Anton Churyumov. Byteball: A decentralized system for storage and transfer of value, 2016. URL: `https://byteball.org/Byteball.pdf`.

10   Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125. Springer, 2016.

11   Andrew Cullen, Pietro Ferraro, Christopher King, and Robert Shorten. Distributed ledger technology for iot: Parasite chain attacks. *CoRR*, 2019.

12   Ali Dorri, Salil S Kanhere, and Raja Jurdak. Towards an optimized blockchain for iot. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, pages 173–178. ACM, 2017.

13   Tom Everitt and Marcus Hutter. A Topological Approach to Meta-heuristics : Analytical Results on the BFS vs . DFS Algorithm Selection Problem . *CoRR*, pages 1–58, 2018. `arXiv:arXiv:1509.02709v2`.

14   Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 51–68. ACM, 2017.

15   Huberman Gur, Leshno Jacob D., and Moallemi Ciamac C. Monopoly without a Monopolist: An Economic Analysis of the Bitcoin Payment System, 2017.

16   K. Karlsson, W. Jiang, S. Wicker, D. Adams, E. Ma, R. van Renesse, and H. Weatherspoon. Vegvisir: A Partition-Tolerant Blockchain for the Internet-of-Things. In *Proceedings of the IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 1150–1158, July 2018. `doi:10.1109/ICDCS.2018.00114`.

17   Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, 2017.

18   Bartosz Kuśmierz and Alon Gal. Probability of being left behind and probability of becoming permanent tip in the Tangle, 2016.

19   Yoad Lewenberg, Yonatan Sompolinsky, and Aviv Zohar. Inclusive block chain protocols. In *Proceedings of the 19th International Conference on Financial Cryptography and Data Security*, pages 528–547. Springer, 2015.

20 Chenxing Li, Peilun Li, Dong Zhou, Wei Xu, Fan Long, and Andrew Yao. Scaling Nakamoto Consensus to Thousands of Transactions per Second. *ArXiv e-prints*, page arXiv:1805.03870, May 2018. `arXiv:1805.03870`.

21 Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, August 2017. `doi:10.1016/j.future.2017.08.020`.

22 Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 17–30. ACM, 2016.

23 Ujan Mukhopadhyay, Anthony Skjellum, Oluwakemi Hambolu, Jon Oakley, Lu Yu, and Richard Brooks. A brief survey of cryptocurrency systems. In *Proceedings of the 14th Annual Conference on Privacy, Security and Trust (PST)*, pages 745–752. IEEE, 2016.

24 Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. URL: `https://bitcoin.org/bitcoin.pdf`.

25 Ferraro P., King C., and Shorten R. Iota-based directed acyclic graphs without orphans, 2019.

26 Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, 2016. URL: `https://lightning.network/lightning-network-paper.pdf`.

27 Serguei Popov. The tangle, 2015. URL: `https://iota.org/IOTA_Whitepaper.pdf`.

28 Meni Rosenfeld. Analysis of hashrate-based double-spending. *CoRR*, pages 1–13, 2014. `arXiv:arXiv:1402.2009v1`.

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