Efficient and Secure TSA for the Tangle

Quentin Bramas

Efficient and Secure TSA for the Tangle

Quentin Bramas

ICUBE, Strasbourg University, CNRS, France

Abstract

The Tangle is the data structure used to store transactions in the IOTA cryptocurrency. In the Tangle, each block has two parents. As a result, the blocks do not form a chain, but a directed acyclic graph. In traditional Blockchain, a new block is appended to the heaviest chain in case of fork. In the Tangle, the parent selection is done by the Tip Selection Algorithm (TSA). In this paper, we make some important observations about the security of existing TSAs. We then propose a new TSA that has low complexity and is more secure than previous TSAs.

1 Introduction and Background

A Distributed Ledger Technology (DLT) is a distributed protocol executed by a set of nodes to maintain an append-only data structure. In Bitcoin, the data-structure is a chain of blocks, containing transactions. Blocks are appended one after the other to form a chain. Each block requires some amount of computational power, *called weight*, to be created. In Bitcoin (and other Proof-of-Work Blockchains), a new block is added to the heaviest branch *i.e.*, the branch that maximizes the sum of the weights of the blocks it contains. This behavior is at the core of the security of Bitcoin.

In this brief announcement, we are interested in the data structure called the *Tangle*, used to store transactions in the IOTA cryptocurrency, and especially in the algorithm used to append new data. We make some important observations about the security of such algorithms and how previous algorithms do not satisfy them. We then propose a new algorithm that is more secure and more efficient than previous solutions.

The Tangle. *The Tangle* is a data-structure where each block of transactions, called *site*, is linked to two previous sites (using hash pointers), called *parents*. The *genesis* site is the only site without parents. Thus, sites form a Directed Acyclic Graph (DAG) of sites. A site is said to *confirm* all its ancestors in the Tangle. A *tip* of the Tangle is a site which has no child *i.e.*, which is not confirmed by any site.

We consider a network composed of connected nodes that generate and broadcast new sites. Each node has a local copy of the tangle that is updated when a new site is appended.

In order to append a transaction in the Tangle, a node must perform a Proof-of-Work *i.e.*, solving a cryptographic puzzle requiring a certain amount of computational power. The *weight* of a site represents this work and we assume each site has a weight of 1. Then, the *cumulative weight* of a site is defined [6] as the sum of its own weight with the weight of its descendants (the sites that confirm it).

Tips Selection Algorithm (TSA). When a site is added to the Tangle, its parents are selected by a *Tip Selection Algorithm* (TSA). The TSA must select two tips (unconfirmed sites) that are not conflicting (informally, two transactions are conflicting if accepting both would produce a double spend). The TSA is a fundamental component of the protocol because it implicitly indicates how the nodes agree on the current state of the Tangle. Indeed, if two tips are conflicting, the TSA indicates which one is considered correct (and should be extended by appending a new site to it) or orphaned (by ignoring it).

Since each node in the network maintains its own version of the Tangle, a site can end up having multiple children. Indeed, due to the latency in the network, the TSA could chose a site which is a tip locally, but that is already confirmed in another version. The Tangle whitepaper [6] presents two TSAs[1]:

[1] A third one is briefly presented but is actually just a variation of the MCMC that we present here.

- Uniform TSA: Each parent is chosen uniformly at random among all the tips.
- Markov Chain Monte Carlo (MCMC): the selection of each parent is done by using a random walk. A walker starts from a given site (eg, the genesis), moves from site to child site, and stops when it reaches a tip. The probability of moving to a child site, depends on its cumulative weight (see [6] for more details).

The Double Spending Attack. In the Tangle, an attacker that wants to double spend must generate two conflicting transactions and first broadcast only one of them. When the first transaction is considered well-confirmed (*i.e.*, the honest nodes think the probability to reverse it is small enough), then the attacker can broadcast the second transaction and append a lots of sites (forming *a parasite chain* [6, 2]) so that the first transaction is discarded.

2 Related Work and Motivations

Related Work. The Uniform TSA was initially proposed for its simplicity. One of its advantages is that tips are quickly confirmed [6, 5]. However, it is easy to see that it offers no protection against double spending attacks. Indeed, an attacker just has to generate more tips than the current number of honest tips to have a higher probability to be selected by honest nodes. Hence, even very old transactions could be canceled easily.

The MCMC algorithm was the first to offer protection against double spending attacks. Indeed, the older a transaction is, the harder it is to cancel it [6, 1]. However, the MCMC requires computing the cumulative weight of every sites in the tangle (which has worst-case quadratic complexity in the number of sites), and its security depends on a parameter α which also influences the number of tips that are left behind [5] (*i.e.*, tips that are never confirmed). In other words, better security implies less stability, and usability.

The efficiency and the security of the MCMC has been improved with MCMC$_{rw}$ [1], by using a simpler version of the cumulative weight (which has linear complexity in the number of tips). MCMC$_{rw}$ obtains a better trade-off security/stability than standard MCMC. G-IOTA [3] and E-IOTA [4] are two extensions of IOTA that proposed mechanisms to limit the number of left-behind tips, while still using MCMC for its security.

All the previously proposed TSAs mixes in the same algorithm the security and the stability aspects. Our goal is to give an algorithm that separates these two aspects.

The last version of the IOTA whitepaper [7] has a similar approach. It proposes to use a completely distinct algorithm to resolve conflicts so that the TSA is not concerned by the security aspect. However, the security of the proposed consensus algorithm has not yet been formally studied. Our goal is to improve previously defined TSAs, using the same model as the original Tangle whitepaper, which has been formally defined [6, 2].

Motivations. Our motivation comes from three important observations.

Observation 1. *If, between two conflicting transactions, one is considered malicious*[2] *with higher probability than the other, does it make sense to choose the malicious transaction as parent with non-zero probability ?*

Regardless of the algorithm used to compare conflicting transactions, we believe a transaction that is considered malicious should never be selected as parent, even with small probability. Otherwise, a fraction of the honest nodes will support the malicious transactions and help the adversary. So, we think a secure TSA should resolves conflicts in a **deterministic** manner, using another algorithm that we call the *Conflict Resolving Algorithm* (CRA).

Observation 2. *The uniform random tip selection is the algorithm that offers the best confirmation time and produces the smallest number of tips on average. However it offers poor security guarantees.*

[2]Here malicious just means that it conflicts with a transaction that is considered correct

The main reason Uniform random TSA is not used in practice is because it offers poor security guarantees. Indeed, it is very easy for a malicious node to generate a small number of transactions to give a high probability for an old transaction to be selected as parent. However, when there is no conflicts, transactions are confirmed very quickly and no transaction is left over. Thus, there is no issue in using the uniform random TSA, after that the set of non-conflicting tips has been deterministically selected.

Observation 3. *MCMC offers good security guarantees at the price of slower confirmation time and higher number of tips on average.*

Again, if an algorithm provides a good way to discriminate conflicting transactions, then there in no reason not to use it for this purpose. Then, another algorithm can be used to randomly select parents among the non-conflicting remaining tips efficiently. The security of MCMC is due to the fact that a random walker has a greater probability to move towards sites with higher cumulative weight. However, we think there is no need to do it for all the sites, but instead, it should be done only when comparing conflicting sites.

3 A New Secure TSA: the two-step TSA

Model. Given a Tangle, $\mathcal{S}$ denotes the set of sites. For any subset C of sites, we say that C is *conflict-free* if all the sites in C are pairwise non-conflicting. We now give a more precise definition of tips that takes into account conflicts. We say a conflict-free set C is a *set of tips*, if there is no sites $s \in \mathcal{S}$ and $t \in C$ such that s confirms t and $C \cup \{s\}$ is conflict-free. This means that, if a tip in C is confirmed by some site in $s \in \mathcal{S}$, then s does conflict with another site in C. For a site s, $w(s)$ denotes its cumulative weight.

The 2-Steps TSA. Our 2-Step TSA first resolves conflicts between sites and then dispatch parents among conflict-free sites.

Our Conflict Resolver Algorithm (CRA) takes a Tangle and returns a maximal conflict-free set of tips C such that, for any pair of conflicting sites s_1 and s_2, if s_1 is confirmed by some site in C, then $w(s_1) \geq w(s_2)$ *i.e.*, the conflict-free set of tips that confirms only the heaviest site in case of conflicts, and is maximal in the sense that no more site can be added to the set without creating conflicts.

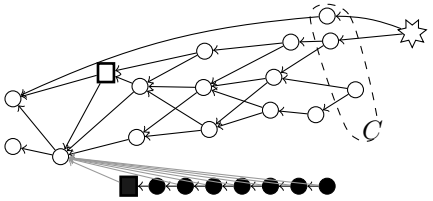

Figure 1: In this example, the white square is considered correct and the black one is discarded.

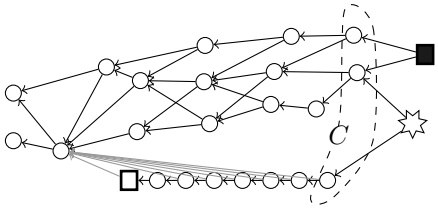

Figure 2: In this example, the new site can merge both branches.

Our Tip Dispatcher Algorithm (TDA) takes a set of conflict-free tips C and returns two tips p_1 and p_2 selected uniformly at random among C, with p_1 and p_2 distinct if $|C| \geq 2$.

In Figure 1 we see a tangle and two conflicting transactions (the two squares). Our CRA first discriminates between the two and considers the white square to be correct and discards the black sites (all the sites confirming the black square). The output of the CRA is the set C containing three conflict-free tips. Then our TDA dispatches the two parents without discriminating between the old and the recent sites. The goal of the TDA is to confirm as many sites as possible, reducing the number of left-over sites.

Security. Using our algorithm, if the honest nodes agree on a conflict-free set of tips, then they all extend the tangle in the same way, increasing the weight of the same set of sites. In other words, for any discarded site, there is a site, considered correct, whose weight increases for each new honest site.

This implies that, if an adversary wants to discard a site that is considered correct, it has to generate sites at a higher rate than the honest nodes (which is a necessary assumption anyway [2]). It means that, like in Bitcoin, the probability of creating a successful double spending attack on a site decreases exponentially fast with its weight.

This property is not obtained by previous TSAs. For instance, if a parasite chain has probability 1/3 to be selected by the MCMC, then an honest node will append one third of its transactions in the parasite chain (assuming they are independent), which is not the intended behavior. In addition, 1/3 of the honest transactions globally will end up selecting the parasite chain as the correct one. A third of the honest nodes plus the malicious node then represents half of the computational power, so that it becomes even easier for the malicious node to increase the probability of selecting its parasite chain.

Using our TSA, the parasite chain is never selected if its cumulative weight is smaller than another branch of the tangle. This implies that a malicious node that wants to double spend has to create a parasite chain on its own and is not helped by honest nodes.

Another interesting property of our TSA is that it does not automatically consider correct a site that is located on the main branch. Instead, it compares conflicting sites independently on where they are located on the tangle. Doing so, we can confirm a separate branch that could look like a parasite chain, but is in fact older and might contain honest transactions as well, for instance if it was generated offline. Indeed, we do not want to discard an entire chain just because a conflicting site appears on top of the main chain. Figure 2 illustrates the situation. We see that the white square has a greater cumulative weight compared to the black square, so only the site confirming the black square (there is no such site in this example) are discarded, creating two tips (the parents of the black-square site). We then have a chance to merge the two branches with a new site (the star-shaped one) using our TDA.

In this situation the MCMC would choose the main branch with greater probability and would almost never merge both branches since the MCMC would never stops its random walk to a parent of black square because it is not a tip.

Performances. Despite using the cumulative weight, which is computed in $\Theta(n)$ time for a given site, our algorithm can have constant complexity in most situations.

After receiving the Tangle from its peers, a node can compute the conflict-free set of tips C with the CRA, while storing the cumulative weight of each site for later use. After that, every time the node has to generate a site s, the TSA will return two parents p_1 and p_2 among C and there is no need to run the CRA again for the next site as the new conflict-free set of tips is simply $C \cup \{s\} \setminus \{p_1, p_2\}$. Similarly, for each incoming site s, if s confirms a site in C, then we know s is considered correct and we can update C by adding s and removing the confirmed tips. So if all the nodes are honest, after the first run of the CRA, every execution of the TSA has constant-time complexity.

However, if an incoming site confirms a site s_m, considered malicious, and conflicting a site s_c considered correct, then we can increment the weight of s_m by one and compare it to the previously computed weight of s_c. If the weight of s_m is still smaller than the weight of s_c, we can safely ignore the new site as running the CRA again will not change our current conflict-free set of sites C. If the weight of s_m becomes greater than the previously computed weight of s_c, then we have to update the weight of s_c and do the comparison again. We believe other optimizations could be performed in this case as well.

Concluding Remarks. We propose a new paradigm for constructing secure and efficient TSAs. We observed that existing TSAs can be improved by spliting the parent selection into a conflict resolving phase and tip dispatcher phase. We believe this work will open new research on the security and the performances of TSAs.

References

[1] Attias, V., Bramas, Q.: How to choose its parents in the tangle. In: International Conference on Networked Systems. pp. 275–280. Springer (2019)

[2] Bramas, Q.: The Stability and the Security of the Tangle (Apr 2018), `https://hal.archives-ouvertes.fr/hal-01716111`, working paper or preprint

[3] Bu, G., Gürcan, Ö., Potop-Butucaru, M.: G-iota: Fair and confidence aware tangle. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 644–649. IEEE (2019)

[4] Bu, G., Hana, W., Potop-Butucaru, M.: Metamorphic iota. arXiv preprint arXiv:1907.03628 (2019)

[5] Kusmierz, B., Sanders, W., Penzkofer, A., Capossele, A., Gal, A.: Properties of the tangle for uniform random and random walk tip selection. In: 2019 IEEE International Conference on Blockchain (Blockchain). pp. 228–236. IEEE (2019)

[6] Popov, S.: The tangle. white paper (2016), https://iota.org/IOTA_Whitepaper.pdf

[7] Popov, S., Moog, H., Camargo, D., Capossele, A., Dimitrov, V., Gal, A., Greve, A., Kusmierz, B., Mueller, S., Penzkofer, A., et al.: The coordicide (2020)