# Impact of Security Aspects at the IOTA Protocol

Tomáš Janečko(✉) and Ivan Zelinka

Department of Computer Science, VŠB - Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava - Poruba, Czech Republic
{tomas.janecko,ivan.zelinka}@vsb.cz

**Abstract.** This paper presents an impact of security aspects at the IOTA protocol. Based on the different usage of computational resources and the difficulty of the selected Proof of Work (PoW) algorithms have been explored the consequences in the final behavior of the IOTA network and the throughput of the protocol implementation. The main feature of the IOTA is the *tangle* which is the name for the directed acyclic graph (DAG). This graph is highly responsible for persisting transactions in the network. The goal of this network is to provide peer-to-peer communication between machines, humans and as well for the Internet of Things (IoT) industry.

**Keywords:** IOTA · Proof of work · Performance · Security · IoT

## 1 Introduction

With the advent of distributed ledger technologies, we are now able to distribute and synchronize ledgers of data and money in secure, distributed, decentralized and permissionless environments. By removing the need for trusted third-parties as the gatekeepers and arbiters of truth, enormous efficiency gains, innovation opportunities and new value propositions emerge [1].

In the increasing adoption of IoT technologies and industry is necessary to provide any system which will be able to handle massive load of microtransactions and data integrity for machines. For that was created cryptocurrency called IOTA which is the main token in the current network. To master this the [1] IOTA's distributed ledger does not consist of transactions grouped into blocks and stored in sequential chains, but as a stream of individual transactions entangled together.

## 2 The Tangle

The Tangle, which is the data structure behind IOTA, is a particular kind of directed graph, which holds transactions. Each transaction is represented as a vertex in the graph. When a new transaction joins the tangle, it chooses two


A. Abraham et al. (Eds.): IITI 2018, AISC 875, pp. 41–48, 2019.
https://doi.org/10.1007/978-3-030-01821-4_5

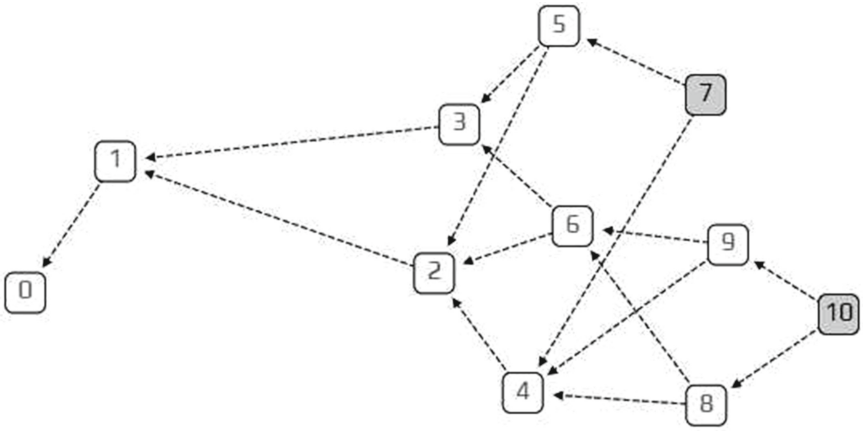


**Fig. 1.** The Tangle overview

previous transactions to approve, adding two new edges to the graph [1]. The example of the graph is presented in Fig. 1.

In the example above transaction number 6 approves transactions number 2 and 3. The greyed out transactions number 7 and 10 are called *tips*. The tips are the unconfirmed transactions in the tangle graph because no one approved it yet. Each new incoming transaction needs to choose two tips to approve. The key point for the correct behavior of the network is to choose the correct algorithm for tips selection.

## 3 Experiment Design

In our testnet environment we have been observing how exactly the transactions are behaving in the IOTA network depending on the different Proof of Work (PoW) algorithm implementations and their selected difficulties. Because as we know from the real world scenarios, each node uses different strategy and that various implementations have direct impact on the whole Tangle and the organic growth of the network.

Making a transaction in the IOTA network is 3 step process which needs to be completed before the transaction is propagated into the network [4]:

- Signing - Your node creates a transaction and sign it with your private key
- Tip Selection - Your node chooses two other unconfirmed transactions (tips) using the Random Walk Monte Carlo (RWMC) algorithm
- Proof of Work - Your node checks if the two transactions are not conflicting. Next, the node must do some Proof of Work (PoW) by solving a cryptographic puzzle (hashcash)

Hashcash works by repeatedly hashing the same data with a tiny variation until a hash is found with a certain number of leading zero bits. This PoW is

to prevent spam and Sybil attacks. A Sybil attack is based on the assumption, that half of all hash power is coming from malicious nodes [4].

An important property of the hashcash puzzle (and all proof-of-work puzzles) is that they are very expensive to solve, but it is comparatively cheap to verify the solution [3].

### 3.1 Trinary Numeral System

For the computational purposes and outputs IOTA uses the trinary numeral system. This system can have two types:

- Balanced trinary system - Trit can have values −1, 0, 1
- Unbalanced trinary system - Trit can have values 0, 1, 2

Trit unit is analogous to the bit and means Trinary Digit. Tryte means Trinary Byte and is anologous to byte where Tryte consists of 3 bits.

A tryte has 3 trits, so the maximum value will be $(3^3 - 1)/2 = 13$ and it has $3^3 = 27$ combinations. This is caused because the values in the trinary system are balanced around the zero.

For the IOTA purposes and the better human readability have been created IOTA tryte alphabet. The tryte alphabet consists of 26 letters of the latin alphabet plus the number 9 and the tryte alphabet has a total of 27 characters. Because 1 tryte has $3^3 = 27$ combinations, each tryte can be represented by a character in the tryte alphabet: 9ABCDEFGHIJKLMNOPQRSTUVWXYZ [4]. The alphabet is visible in the Table 1.

**Table 1.** IOTA tryte alphabet

| Tryte | Decimal | Char | Tryte | Decimal | Char |
|---|---|---|---|---|---|
| 0, 0, 0 | 0 | 9 | | | |
| 1, 0, 0 | 1 | A | −1, −1, −1 | −13 | N |
| −1, 1, 0 | 2 | B | 0, −1, −1 | −12 | O |
| 0, 1, 0 | 3 | C | 1, −1, −1 | −11 | P |
| 1, 1, 0 | 4 | D | −1, 0, −1 | −10 | Q |
| −1, −1, 1 | 5 | E | 0, 0, −1 | −9 | R |
| 0, −1, 1 | 6 | F | 1, 0, −1 | −8 | S |
| 1, −1, 1 | 7 | G | −1, 1, −1 | −7 | T |
| −1, 0, 1 | 8 | H | 0, 1, −1 | −6 | U |
| 0, 0, 1 | 9 | I | 1, 1, −1 | −5 | V |
| 1, 0, 1 | 10 | J | −1, −1, 0 | −4 | W |
| −1, 1, 1 | 11 | K | 0, −1, 0 | −3 | X |
| 0, 1, 1 | 12 | L | 1, −1, 0 | −2 | Y |
| 1, 1, 1 | 13 | M | −1, 0, 0 | −1 | Z |

### 3.2 Minimum Weight Magnitude

The difficulty of the PoW is set by a variable called Minimum Weight Magnitude (MWM). This refers to the number of trailing zeros (in trits) in transaction hash. MWM is proportional to the difficulty of the Proof of Work [5].

The device which does the PoW will bruteforce the transaction hash to find a nonce that, hashed together with the transaction's trits, will result in a transaction hash that has the correct number of trailing 0's. Every extra trailing zero to be found will increase the difficulty of PoW by 3 times [5].

The currently applied parameters are as follows for the IOTA reference implementation (IRI) [4]:

- On the mainnet the minWeightMagnitude = 14 (Applies to IRI release: v1.4.1.2)
- On the testnet the minWeightMagnitude = 9 (Applies to IRI release: testnet-v1.4.1.2)

Higher minWeightMagnitude values should be no problem but will just cause the Proof of Work to take longer unnecessarily. The other side effect is that this longer time will make transactions temporarily invisible for the rest of the transactions and the network could end in the single chain output.

A simplified explanation how hashcash works is as follows. Lets assume that MWM = 3:

- hash(transaction data + counter) = . . . 704c19cddf95 (PoW is not completed)
- hash(transaction data + counter) = . . . 721b564b9000 (PoW is completed)

### 3.3 Quantum Computers Resistance Transactions

As of 2018, the development of actual quantum computers is still in its infancy, but experiments have been carried out in which quantum computational operations were executed on a very small number of quantum bits [6].

Large-scale quantum computers would theoretically be able to solve certain problems much more quickly than any classical computers that use even the best currently known algorithms [7].

The goal of the IOTA network and the experiment is to be quantum resistant. To achieve that, the solution for the PoW and hash computations, need to be done by One Time Signatures (OTS) cryptography.

The term implies that a single public/private key pair must only be used once. Otherwise, an attacker is able to reveal more parts of the private key and spoof signatures [4].

For the IOTA network was choosed the Winternitz One Time Signature (WOTS) scheme. The WOTS is most suitable for combining it with Merkle's tree authentication scheme because of the small verification key size and the flexible trade-off between signature size and signature generation time. Further

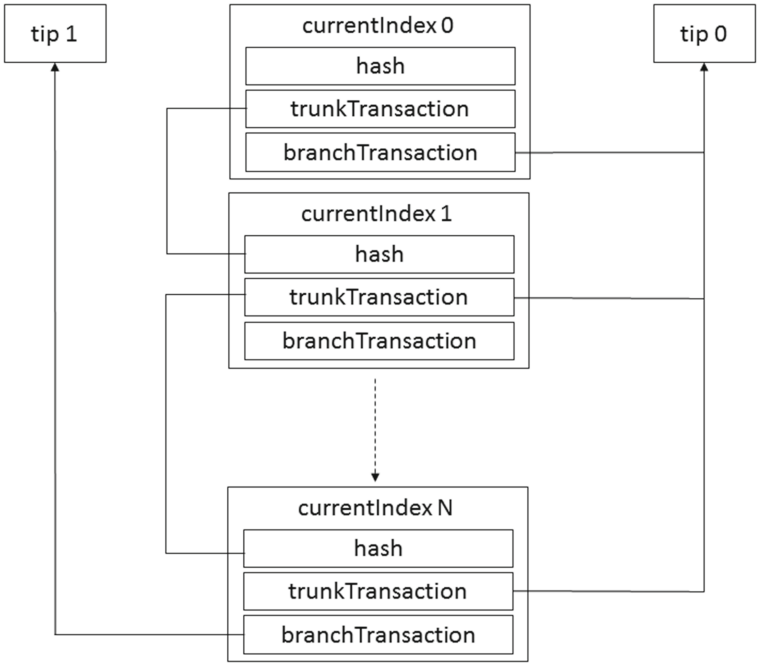


**Fig. 2.** Bundle of transactions

it is possible to compute the corresponding verification key given a W-OTS signature. So a Merkle signature scheme does not need to contain the verification key [8].

For the performance optimization purposes was choosed the solution where the transactions are bundled together, because upon that the signature needs to be passed into each transaction. The structure of the bundle is shown at Fig. 2 [4].

All transactions in the same bundle should be treated as an atomic unit. It means that either all transactions of a bundle are confirmed, or none of them are confirmed and every transaction in the bundle requires it own PoW [4], so they have the different nonces in the transactions.

For the transactions we have used 3 different security levels that impacted the final size of the transaction.

- Security level 1 - The signature is stored in 1 transaction
- Security level 2 - The signature is stored and partitioned into 2 transactions
- Security level 3 - The signature is stored and partitioned into 3 transactions

By increasing the security level we increased the signature size and thus the number of transactions needed to store the signature and that leads to that IOTA signatures are larger than Bitcoin signatures due to IOTA's use of Winternitz one-time signatures to gain quantum resistance.

As an output we have seen that each single transaction inside a bundle consists of 2673 trytes and much of it is taken by the signature message fragment which has a size of 2187 trytes which is approx. 82% [4].

Convert trytes to bytes:

$$bytes = \frac{trytes \times 3 \times ln(3)}{ln(2)}/8 \tag{1}$$

The final size of a single transaction inside a bundle requires 2673 trytes or 1.55 kB.

### 3.4 Proof of Work

The key point in the building of transactions is the PoW mechanism. To be fully comply with the IOTA protocol, the transaction consists of 3 steps:

**Constructing the Bundle.** Constructing the bundle and signing the transaction inputs with your private keys. IOTA uses a bundle which consists of multiple transactions containing credits to the receiving addresses (outputs) and debits from the spending addresses (inputs). In IOTA there are two types of transactions: one where you transfer value and thus, have to sign inputs, and ones where you simply send a transaction to an address with no value transfer (e.g. a message) [4].

**Tip Selection.** The tip selection is a process whereby you traverse the tangle in a random walk to randomly chose two transactions which will be validated by your transaction. Your transaction checks for example if the descendants of that transaction is valid. If these transactions are valid they will be added to your bundle construct and are called branchTransaction and trunkTransaction [4].

**Computing Hashcash.** Once the bundle is constructed, signed and the tips are added to the bundle, the PoW has to be done for each transaction in the bundle. Every transaction in a bundle requires a nonce (this is the result of the PoW) in order to be accepted by the tangle network [4].

To meet these conditions have been implemented 2 different versions of the hash algorithm of the PoW:

- Webgl 2 Curl implemention - WebGL uses the system Graphics Processing Unit (GPU)
- CCurl implementation - CCurl means C port of the Curl library, which uses the system Central Processing Unit (CPU) (Fig. 3)

During the gathering of the metrics was observed that Webgl 2 Curl implementation executes PoW faster, but method will not work for all users due to the incompatibility by the GPUs. Otherwise CCurl implementation will always work for all users.

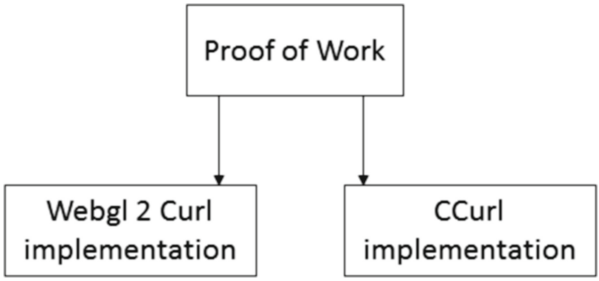


**Fig. 3.** Proof of work

These results leads to that we use the CCurl implementation as a fallback function whether the more efficient Webgl 2 Curl implementation isn't sufficient or compatible.

For the testing purposes as the single node was used machine with specification Intel Core i5-7300 2.7 GHz with 16GB RAM together with Intel HD Graphics 620.

## 4 Results

To uncover the nature of the security in the network, it was necessary to analyze the differences between GPU and CPU implementation of PoW.

Upon closer measurement, it was found that the nodes which are responsible for hashcash calculation and transactions propagation into the network are network creators and determines the direction of the organic growth of the IOTA which has big impact on the whole network.

We distinguish between two regimes, single transaction and bundle. There is only one hash calculation in the single transaction regime and a single transaction can obviously contain multiple inputs and outputs [1].

In the bundle regime IOTA uses an account-like scheme. This means that we have inputs (addresses) which you have to spend in order to transfer tokens. Addresses are generated from private keys, which in turn are derived from a trytes-encoded seed. A transfer in IOTA is a bundle consisting of outputs and inputs. Bundles are atomic transfers, meaning that either all transactions inside the bundle will be accepted by the network, or none. A typical transfer in IOTA is a bundle consisting of 4 transactions [1].

A unique feature of bundles is that the transactions are identified via the bundle hash, but also via the trunkTransaction. What this means is that the tail transaction (currentIndex: 0), references in the trunkTransaction the transaction hash at index: 1, currentIndex 1 transaction references (and approves) index 2 and so on. This makes it possible to get the full bundle of transactions from just a tail transaction by traversing down the trunk transaction [1].

## 5 Conclusion

In this paper, IOTA network was used for investigation of the impact of the security features at the IOTA protocol. From the above findings, we came to the conclusion that the main advantage over other networks is the preparation for resistance to the quantum computers. But the same drawback for that feature is that for each transaction we need to generate new pair of private/public key.

This leads to the weak point of the whole network from the users perspective, because any incorrect usage of the transactions could lead to loss of control over own address balance.

For further research we would like to focus how to apply evolution algorithms which could help with more stable transactions distribution and propagation into the tangle.

The next path for research leads towards to improve nodes scalability to be more agile and be able quickly respond to the changes in the nonce (minimum weight magnitude) setup of the particular network.

**Acknowledgement.** The following grants are acknowledged for the financial support provided for this research by Grant of SGS No. 2018/177, VSB-Technical University of Ostrava and under the support of NAVY and MERLIN research lab.

## References

1. IOTA Foundation. https://www.iota.org
2. Popov, S.: The Tangle. https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf. Accessed 26 June 2018
3. Hashcash.org: Hashcash proof-of-work paper. http://www.hashcash.org/papers/proof-work.pdf. Accessed 2 July 2018
4. Mobilefish. https://www.mobilefish.com. Accessed 2 July 2018
5. IOTA documentation. https://docs.iota.org. Accessed 2 July 2018
6. Gershon, E.: New Qubit control bodes well for future of quantum computing. Phys.org. Accessed 26 Oct 2014
7. Simon, D.R.: On the power of quantum computation. In: Proceedings of 35th Annual Symposium on Foundations of Computer Science, pp. 116–123. CiteSeerX 10.1.1.655.4355 Freely accessible (1994). https://doi.org/10.1109/SFCS.1994.365701. ISBN 0-8186-6580-7
8. Buchmann, J., Dahmen, E., Ereth, S., Hülsing, A., Rückert, M.: On the Security of the Winternitz one-time signature scheme. In: Nitaj, A., Pointcheval, D. (eds.) Progress in Cryptology - AFRICACRYPT 2011. AFRICACRYPT 2011. Lecture Notes in Computer Science, vol. 6737. Springer, Heidelberg (2011)