IOTA Vulnerability: Large Weight Attack Performed in a Network

Lucas de Vries
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
l.j.w.devries@student.utwente.nl

ABSTRACT

IOTA is a public distributed ledger based on a Directed Acyclic Graph called the Tangle. It does not use blocks and a chain like usually used in cryptocurrencies, and consequently, no miners are needed. It has multiple advantages above other blockchains (Bitcoin), amongst others, it is more scalable and performing a transaction involves no costs apart from some computation power. This makes the IOTA revolutionary but there are also known vulnerabilities. One of these vulnerabilities is a large weight attack. In this attack, the attacker performs two fraudulent transactions in which its money is spent twice. One transaction will be validated by the network and the double spending transaction normally will not. However, the attacker just tries to validate this transaction by himself by creating a lot of transactions confirming the double spending transaction. The goal of this research is to demonstrate this attack in a real network and to determine in which scenarios a large weight attack can be successful.

Keywords

IOTA, large weight attack, vulnerability

1. INTRODUCTION

IOTA is aiming to become the underlying architecture, the backbone for the Internet of Things (IoT). [3].

Characteristics that make IOTA suitable to become the backbone for IoT devices:

- IOTA is a cryptocurrency, it has a decentralized distributed ledger,
- it supports microtransactions and data integrity,
- is scalable so it can support the millions of devices,
- it has no transaction costs so it is cheap,
- it does not require much computation power so it can be run on a small IoT device.

Security implications will be catastrophic since it could probably affect millions of devices. Therefore, research on security vulnerabilities is essential. Currently, there are a few vulnerabilities known [9]. Amongst others, a large weight attack can possibly be performed which makes it possible for an attacker to get two conflicting transactions confirmed. Once both the transactions have status 'confirmed', the attack can be called successful.

1.1 Large weight attack

The approach of an attacker in a large weight attack involves a few steps [9, p. 15-16]:

1. The attacker buys something by issuing a legitimate transaction and waits till the goods arrive after the legitimate transaction gets confirmed.
2. The attacker performs a double spending transaction resulting in a new branch.
3. The attacker increases the weight of the double spending transaction by creating a lot of transactions approving the double spending transaction. Consequently, the new branch increases in weight.
4. The network notice that the branch containing the double has a higher weight than the branch containing the legitimate transaction. The branch containing the double spending transaction will become the new main branch and the other branch will be repelled.

Once the network decides to continue with the branch containing the double spending transaction, the attacker is able to spend his funds in both branches.

In this research, a large weight attack will be demonstrated on an online network, namely, the network provided by IOTA for developing [4]. Furthermore, it will be determined how the transaction speed of a network determines the chance on a successful large weight attack.

2. BACKGROUND

2.1 IOTA

D. Sønstebø, S. Ivancheglo, D. Schiener, and S. Popov founded IOTA in 2015[8]. Two years later, the IOTA whitepaper was published by Popov [9]. In that paper, the mathematical foundations of IOTA are analyzed and possible attacks are discussed.

The distributed ledger of IOTA is not composed of blocks and a chain, like cryptos usually do, but a Directed Acyclic Graph (called tangle) is used instead.

2.2 Tangle

The tangle starts with an initial transaction called Genesis. Adding a transaction to the network requires the verification of two previous transactions. This way, each

30th Twente Student Conference on IT Febr. 1st, 2019, Enschede, The Netherlands.

transaction points to the two other transactions and in the graph, these pointers are seen as edges and the transaction as a node. An example of a tangle can be found in Figure 1.

An advantage of the tangle is that transactions will be approved faster when more transactions are being performed since each transaction validates two other transactions.

In addition, no miners are necessary. Everybody can add its own transaction to the network. Only two other transactions need to be validated and a small amount of computation power is required to prevent transaction spamming.

2.3 Proof of Work

The small amount of computation power required for each transaction is called Proof of Work (PoW). Each IOTA network has a minimum required PoW and the value determines the minimum length of trailing zeros after hashing the transaction. By changing the value of the nonce field of a transaction, a user is able to increase the number of trailing zeros. It requires computations to determine the right nonce value. The higher the required PoW, the longer it takes to determine the nonce value, the longer it takes to perform a valid transaction.

2.4 Transaction process

Performing a transaction is briefly explained, however, understanding how transactions are performed form the basis of understanding IOTA. Therefore, it will be explained more in-depth.

Performing a transaction involves:

1. Prepare transaction
2. Select two other transactions
3. Sign transaction
4. Calculate Proof of Work
5. Broadcast through the network

When a user tries to perform a transaction, first the transaction will be prepared. This contains filling in most of the field of the transaction object, for example, the amount of the transaction, the receiver address. The next step is selecting two other transactions (see section 2.6 for more detail). The new transaction will validate these two existing transactions. Now the transaction can be signed using the private key of the user. Before the transaction can be broadcast to the network, the PoW needs to be calculated.

After the transaction is broadcast, the user will have to wait till other transactions select this certain transaction for validation. Once enough other transactions directly or indirectly approved the transaction, it will get status 'confirmed'. Enough is determined by the percentage of new unapproved transactions that are (indirectly) pointing to a certain transaction. Normally, when 95% of the new transaction approve a certain transaction, that will be enough to get status 'confirmed'.

2.5 Coordinator

Since IOTA is in its beginning phase, it has not that many users yet, so the transaction speed is currently not very high. The problems that come with a low transaction speed is that it takes long before transactions are approved and it becomes easy for attackers to obtain a large part of the hashing power (required for the PoW). When an attacker has greater than50% of the computation power, it can force which transactions are being confirmed. To solve these two problems, IOTA initialized a so-called Coordinator. This is a trusted central authority which determines which transactions get status 'confirmed'.

2.6 Tip selection

As mentioned, a new transaction needs to validate two other transactions. The way these two transactions are chosen has consequences for the network. If nobody picks a certain transaction, this transaction will never be approved. An easy solution would be to only select tips (transactions that are not yet validated at all) randomly. However, it can occur that someone tries to spend its funds twice and that will split the tangle since no transaction can verify the branch with the legitimate transaction and the branch with the double spending transaction. Eventually, only one branch should survive and the other should be dropped. It is desirable that it is decided which branch will survive early since all the transactions of the other branch will be dropped. In the current IOTA implementation, only transactions that are confirmed by the coordinator will be selected for verification.

3. RELATED WORK

Since IOTA is relatively new, there is not much published research about security vulnerabilities. The IOTA white paper addresses multiple attacking scenarios. In the first attacking scenario, the attacker tries to outpace the rest of the network [9, p. 15-19]. This requires an equal amount of computation power as the rest of the network so that will not be feasible in a larger network. A second scenario describes a so-called parasite attack [9, p. 19-23]. The attacker creates a chain of transactions (parasite). This parasite tries to approve an invalid transaction and because it is a large chain which claims the invalid transaction is valid, the network might be convinced and approve the transaction (very much simplified). Furthermore, a splitting attack is addressed in the same paper [9, p. 23-25]. The attack is described and two countermeasures are suggested. The first suggestion is to use a "sharp threshold", that means that if one branch is slightly larger than the other branch, every new node will come after the largest branch. The second suggestion is having a node publish multiple transactions at once on one of the branches. Consequently, the branches differ in size a lot and it is harder for the attacker to maintain equal size between the branches.

The countermeasures are based on changes in the algorithm which determines where the network grows. That means that if the network is split first and the algorithm determines to only grow at one of the branches, the other branch will vanish.

Heilman et al. published a paper on the cryptographic Curl-P hashing function which was used in IOTA [2]. It has been proven that the Curl-P function was insecure and so it was replaced by the KECCAK hashing function. However, in some parts, the Curl-P function is still being used.

The fundamental research on the IOTA security has been laid, but there is a lot of space for further research. A large weight attack has been researched already, however, it is not tested in a network.

4. PROBLEM STATEMENT

The theory of a large weight attack has been researched and countermeasures are suggested, however, the attack is not yet practically exploited. A large weight attack is not yet performed in a network and therefore, it is unclear how the transactions speed influences the attack. This leads to the following research questions.

4.1 Research Questions

1. How can a large weight attack be performed?
2. How do the size and transaction speed (transactions per second) in a network influence the chances of a successful large weight attack?

5. METHODS

5.1 Architecture

IOTA always runs in a network containing one or more full nodes. A full node contains a database with all transactions (the Tangle). In this research, as a network, the devnet [4] provided by the IOTA Foundation [5] is used. To participate in this network, a connection to the full node, provided by IOTA, is made. The node has an API which is used to send transaction to. The python iota library [6] is used to compose transactions which can be sent to the API.

5.2 Composing transactions

Transactions are composed using the following code:

```
def main(address, message, value):
    api = Iota(uri, seed)
    api.send_transfer(
        depth=3,
        transfers=[
            ProposedTransaction(
                # Recipient of the transfer.
                address=Address(address),
                # Amount of IOTA to transfer.
                value=value,
            ),
        ],
    )
```

First, the API object needs to be created. The PyOTA library provides a constructor with as first argument the URI of the node you want to connect with and with your seed as the second argument. Once the connection to the node is made, a transaction can be composed. The send_transfer method (from the PyOTA library) is used for that and the required parameters are passed to this method. These parameters are the depth, this is relevant for the tip selection, the receivers address and the number of IOTA that needs to be transferred. The send_transfer method results in a bundle containing a number of transactions.

5.3 Changes to the PyOTA library

A transaction has a number of fields containing the information of the transaction. Two of these fields are 'trunkTransaction' and 'branchTransaction' and these are the fields in which the values determine which transactions are approved by this transaction. Increasing the weight of a transaction would require a number of other transactions approving that certain transaction. Therefore, an attacker wants to set these fields manually. In this research, this is implemented by making small changes to the PyOTA library.

Normally, the send_transfer method makes an API call to the full node with which it is connected and asks which transactions it should approve. By changing the return values of the method in which the API call is made, it became possible to set a value manually. The changes are made in the send_trytes.py file on lines 52 and 53 (folder of this file:
iota.lib.py/iota/commands/extended/).

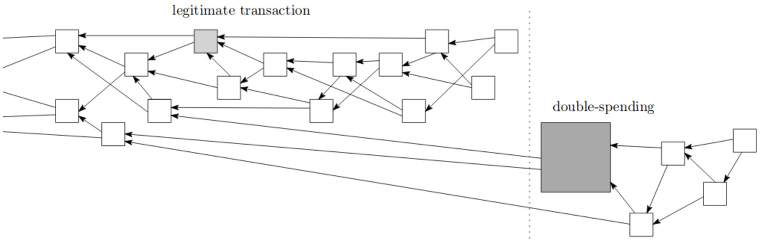

Figure 1. Transaction flow in a large weight attack. Diagram is from the Whitepaper [9].

5.4 Attack approach

The transaction flow of large weight attack is visualized in Figure 1. The attack performed in this research used the same concept as in the figure. So, a legitimate transaction, in which more than half of the funds of a seed are spent, and an equally double spending transaction are performed. Because the PyOTA library has a number of checks before it sends a transaction, these two transactions are performed seconds after each other. This way, the legitimate transaction is not yet processed before the double spending transaction is composed. As shown in figure 1, it is important that the double spending transaction is attached to transactions that are not directly or indirectly improving the legitimate transaction. Otherwise increasing the weight of the double spending transaction would also increase the weight of the legitimate transaction.

After the double spending transaction is sent, its transaction hash is stored. This hash is necessary because the goal is to create new transactions validating the double spending transaction. A script continuously sending zero value transactions with both the 'trunkTransaction' and 'branchTransaction' set to the hash value of the double spending transaction. This script contains a while loop that keeps sending the transactions.

5.5 Device

The device used for the attack was a HP 15bs183nd laptop with an Intel Core i5-8250U and 8GB DDR4 RAM.

6. RESULTS

The most surprising result would be that the double spending transaction was confirmed. Since the coordinator determines which transactions get confirmed, a coordinator confirming a malicious transaction would show that the coordinator did not act to the rules. However, the double spending transaction did not get confirmed.

The double spending transaction bundle can be found at `https://is.gd/doubleSpending` and the legitimate transaction bundle at `https://is.gd/legitimateTransaction`. These links refer to TheTangle.org[10].

6.1 Weight

By creating a lot of transactions validating the double spending transaction, it has been shown that it is possible to manually improve the weight of a malicious transaction. Using the tangle visualizer glumb [1], the screenshot in Figure 2 is created. This figure visualizes the attack.

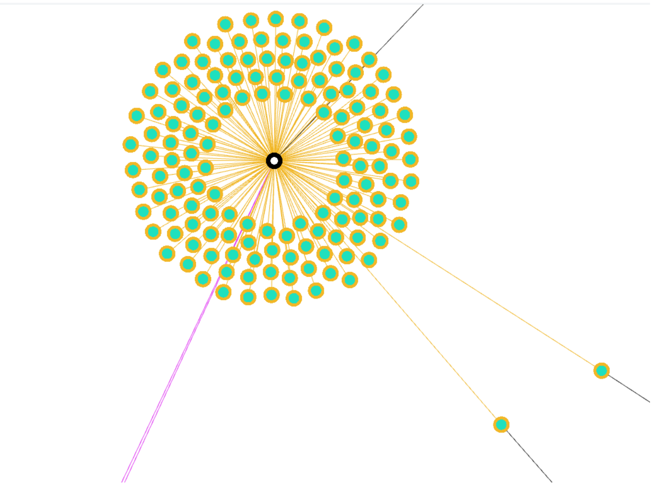

Figure 2. Double spending transaction in the center is being approved by a lot of transactions created by the attacker (blue circles with orange border).

6.2 Transactions per second

The number of transactions created per second declined over time. During a measurement of 52 seconds, 34 transactions were created. That means that a single device is able to issue 0.65 transactions per second (on peak). The legitimate transaction was approved by 1789 other transactions in 1 hour and 23 minutes. That are approximately 0.347 transactions per seconds. This means it is possible for a single device to outpace the network since the speed of the single device is higher than the rest of the network.

After sending around 200 transactions to the API of the full node an API error came up with the following message: ”Too many requests, use your own hardware.” Apparently, the node has built-in protection against spamming and after too many requests it stops handling these requests.

7. DISCUSSION

7.1 Coordinator free network

The double spending transaction did not get approved so the coordinator did not fall for the large weight of the transaction. However, in a coordinator free network, the nodes determine themselves which transactions get the status ’confirmed’. In that case, a larger weight is convincing. It might happen that first the legitimate transaction gets confirmed and the double spending transaction gets the status ’pending’. After increasing the weight of the double spending transaction the legitimate transaction gets the status ’pending’ again and the double spending transaction gets confirmed.

If that happens, that means an attacker is able to buy a product with the legitimate transaction and with the same money, the attacker is able to buy another product with the double spending transaction.

7.2 Transaction speed

In section 6.2 it is shown that it is possible to compose 0.65 transactions per second. This value is depending on the required PoW (see section 2.3) and the computation power of the machine on which the transactions are created.

7.2.1 Proof of Work

The required PoW for Devnet is 9 [4], this is lower than the required PoW of 14 on the Mainnet, which is the network that really is being used [7]. Increasing the required PoW results in a longer computation time per transaction. Therefore, it will become harder to perform a large weight attack. However, a downside on a high PoW is that IoT devices do not have enough computation power to create a transaction in a reasonable time. A possible solution to that is using PoW proxies. These are services that calculate the nonce of a transaction so the IoT devices do not have to do that by themselves.

7.2.2 Using a full node

As described in section 5.1, a connection to a full node is used to send the transactions. Not running your own full node has a few disadvantages related to the transaction speed. The first disadvantage is being dependent on the connection. When the connection becomes slow, the whole process of performing transactions decreases. This has also to do with the implementation that is used. In fact, it would be possible to create new transaction even when the connection to the node is lost for a while. Another drawback is that the API of the full node only accepts a maximum number of transactions in a certain time. As mentioned, after approximately 200 transactions an API error popped up.

7.2.3 Required transaction speed

The minimum required transaction speed to succeed a large weight attack depends on the approach. It is possible to create the double spending transaction and all transactions approving the double spending transaction on an offline subtangle. That means, you create all the transactions but do not yet broadcast it to the network. That means all honest nodes will approve the legitimate transaction and the attacker is the only one approving the double spending transaction. With total computation power X the computation power of the attack a should be larger than $X/2$ so $a > 0,5X$ will result in a larger weight of the double spending transaction than the legitimate transaction.

In the other approach, the attack sends the malicious transactions directly to the network. In that case, other honest nodes have to decide whether they approve the legitimate or the double spending transaction. This will result in a splitting attack where the attack should try to maintain the balance between the branch of the legitimate transaction and the branch of the double spending transaction. How much computation power is necessary to maintain balance is depending on the tip selection algorithm. This algorithm is not yet implemented by IOTA. In the Whitepaper, it is suggested to use the total weight of each of the branches and prefer the branch with the highest weight to select the tips from [9, p. 21]. Once one branch becomes slightly larger than the other, honest nodes will only select tips from the larges branch so again $> 50\%$ of the computation power is required for a successful attack.

7.3 Validation only one transaction

In the attack, the value of the transaction fields ’trunktransaction’ and the ’branchtransaction’ were set to the same transaction hash. That means that by the new transaction only one other transaction is validated. The concept of IOTA is that every transaction validates two other transactions but it has been shown that validation only one transaction also works. If only one user only validates one transaction, this has a negligible effect on the network. However, the moment everybody creates transactions approving only one other transaction, it will have a negative effect on the connectivity of the graph. This will result in a longer validation time. Therefore, users have an incentive to validate two transactions. Individually, only validation one transaction might decrease the time it takes to compose a transaction although the time it takes to calculate the PoW is much longer than validation a transaction.

In conclusion, blocking all transactions only validation one other might not be necessary, however, it also does not have downsides.

8. CONCLUSION

In this research, a large weight attack has been performed in a network. This is done by making changes in the IOTA library for Python. During the preparation of a transaction, the library requests the node, with which it is connected, which two transactions need to be approved. The return value of this method is edited so the value can be set manually.

The large weight attack did not result in a possibility to spend funds twice. A well-functioning coordinator only confirmed the legitimate transaction and not the double spending transaction.

The size of a network does not influence the chance on a successful large weight attack but the transaction speed of the network as a whole does. How this influence the chances of an attacker depends on the approach of the attack. Creating an offline subtangle requires the attacker to have at least more than half of the computation power of the network. In another approach, where the attacker directly broadcasts its transactions, it will result in a so-called splitting attack. The tangle will be split in a branch behind the legitimate transaction and a branch behind the double spending transaction. The computation power an attacker needs in that scenario is depending on the tip selection algorithm. Honest nodes can only add their transactions to one of the branches. If the tip selection algorithm is based on the total weight of the branches, the required computation power comes close to $> 50\%$ since all honest nodes will select tips from the largest branch. However, this requires further research once the tip selection algorithm for a coordinator free network is published.

Increasing the required Proof of Work does also increase the difficulty of performing a successful large weight attack because it requires more computation power per transaction. However, this will also effect honest nodes and that can become problematic for IoT devices.

References

[1] glumb. the tangle. `http://tangle.glumb.de:8080/`. Accessed 2019-01-17.

[2] E. Heilman, N. Narula, G. Tanzer, J. Lovejoy, M. Colavita, M. Virza, and T. Dryja. tangled-curl/vuln-iota.md at master Âů mit-dci/tangled-curl. `https://github.com/mit-dci/tangled-curl/blob/master/vuln-iota.md#2-disclosure-timeline`. Accessed: 2018-11-25.

[3] IOTA. What is iota? | iota. `https://www.iota.org/get-started/what-is-iota`. Accessed: 2018-11-24.

[4] IOTA Foundation. Devnet - developer network - iota docs. `https://docs.iota.org/introduction/networks/devnet`. Accessed 2019-01-19.

[5] IOTA Foundation. The iota foundation | iota. `https://www.iota.org/the-foundation/the-iota-foundation`. Accessed 2019-01-19.

[6] IOTA Foundation. iotaledger/iota.lib.py: Pyota: The iota python api library. `https://github.com/iotaledger/iota.lib.py`. Accessed 2019-01-19.

[7] IOTA Foundation. Pow on the tangle - iota docs. `https://docs.iota.org/introduction/tangle/proof-of-work`. Accessed 2019-01-26.

[8] IOTA Support (NOT affiliated with the IOTA Foundation). Iota support - what is iota? `https://iotasupport.com/whatisiota.shtml`. Accessed: 2019-01-22.

[9] S. Popov. The tangle. 2016.

[10] TheTangle.org. Iota tangle explorer and statistics - thetangle.org. `https://thetangle.org/`. Accessed 2019-01-26.