How to break IOTA heart by replaying?

Gerard de Roode
University College Twente
Enschede, The Netherlands
g.deroode@student.utwente.nl

Ikram Ullah
Pervasive Systems Group, Dep. of Comp. Science University of Twente
Enschede, The Netherlands
i.ullah@utwente.nl

Paul J. M. Havinga
Pervasive Systems Group, Dep. of Comp. Science University of Twente
Enschede, The Netherlands
p.j.m.havinga@utwente.nl

***Abstract*—IOTA is a novel cryptocurrency that uses distributed ledger technology based on directed acyclic graph data structure. Security of cryptocurrencies ought to be scrutinized in order to acquire esteemed security, attain trust, and accomplish indelible adoption. Although IOTA proffer resilient security controls, IOTA security is not yet well explored. Among all the propounded IOTA vulnerabilities that have been identified, we pragmatically exploit replay attack against IOTA. We further analyze the attack to perceive its impact. Attack methodology and proof of concept for the replay attack is presented. Our proposed exploitation methodology is based upon address reuse, while IOTA in default mode does not reuse addresses. Distrust, and privation of balance can be some of the severe impacts of this vulnerability.**

***Index Terms*—IOTA, Blockchain, cryptocurrency, security exploitation, and replay attack.**

I. Introduction

In this paper we formally and practically demonstrate the exploitation of replay attack against IOTA. Internet of Things (IoT) represents the next generation of the Internet, taking a huge leap in its ability to gather, analyze, and distribute data that we can turn into information, knowledge and ultimately making decisions [25]. IoT promises to bring immense value to our lives in the form of increased efficiency, improved health, safety and experience [28]. Smart logistics is considered as one of the typical IoT domains in which significant increase in efficiency and effectiveness is possible [38].

There are many contributing factors to success of IoT, such as: low cost sensors, low power embedded systems, low power communication protocols and data analysis. The number of connected devices is expected to reach 50 billion by 2020 [27] and this will have a huge impact on economic and social life.

IoT encounters many challenges as mentioned in [24] [26] [39]. Security, privacy, trust, and transparency are among the most frequent ones mentioned. Furthermore managing large number of IoT devices with centralized servers, will introduce single point of failure [24]. As mentioned earlier, IoT has overwhelming applications in logistics that can enhance efficiency and quality of service for the transport and logistics industry. In order to fulfill logistics requirements, highly dynamic, distributed and scalable IoT solutions are required. IoT based distributed smart logistic solutions as proposed in [38] [39] can overcome logistics challenges like real-time monitoring, remote maintenance and integration of global supply chain and reduce the power and computation complexity in IoT devices. However, such solutions face transparency, liability, and security challenges, which are crucial in logistics. Decentralized technologies, such as Distributed Ledger Technology (DLT) [29], are considered as suitable alternatives to manage the huge number of connected IoT devices and overcome IoT security challenges. Distributed ledgers use independent nodes to record, share and synchronize transactions or data in their respective electronic ledgers instead of keeping data centralized as in traditional ledgers [29]. With DLT it is possible to achieve more robust security on IoT devices, maintain trust among IoT devices as well as use cryptocurrencies, which are vital for virtual transactions in IoT. Also, DLT can be used for micropayments and smart contracts which are essential in IoT applications such as smart logistics.

As we move away from a centralized architecture to DLT, the importance of effective and secure cryptocurrency proliferates because IoT devices will be required to perform micro-transactions efficiently, securely, and in real-time and without the need of any third party. Some of the technological advantages of DLT over traditional technologies are: *durability* [36], *transparency* [36], *immutability* [36], and *security* [36]. Blockchain and IOTA are among the main DLT that are implemented in major cryptocurrencies.

Our contribution

As the implementation of DLT in many application domains is evolving, more concrete research is required to ensure how well it can withstand the security threats. Since IOTA is relatively a new technology, there are many known and unknown attack vectors which need to be investigated. Although in the related work it is claimed that replay attack is possible, no exploitation methodology

is provided or explored so far. In this work we have exploited replay attack and presented a formal attack methodology. Our contribution includes a methodology to reuse addresses and perform a replay attack, thereby we demonstrate and discuss its feasibility.

II. Blockchain versus IOTA

Blockchain is a data structure which makes it possible to form a public ledger of transactions. Transactions are stored in a chain of blocks. Each block consists of set of transactions. The ledger is shared among a distributed network of devices. Every node can securely modify the ledger without the help of a centralized authority. If any device wants to modify the ledger, the other devices in the network verify the proposed modifications using consensus algorithms. *Proof of Work(PoW)* [37], *Proof of Stake(PoS)* [37], and *Practical Byzantine fault tolerance(PBFT)* [37] are the widely implemented consensus algorithms. Blockchain allows to efficiently manage data by tracking, detailing, and recording every device and its transactions in the network. It uses global peer to peer network to create an open platform that can provide neutrality, reliability, and security. To make sure every device in the network has the same copy of the ledger, the blocks are chained together by including some data of the previous block in the new block. This data will be the hash of the previous block, which is a unique digest of the data in the previous block. The Blockchain can be visualized as shown in Figure 1.

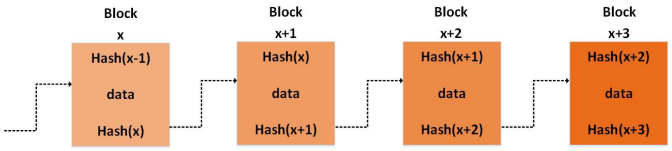

Figure 1. Blockchain visualization, arrows represent links between the blocks

Despite the potential advantages, the Blockchain technology emphatically has some technical challenges [1] [2] [32].

- It is necessary to have miners to secure the blocks and the miners have to be rewarded in some way for their activities. This is done through transaction fees. However, it would be ideal if transactions could be feeless. The importance of micropayments will increase in the rapidly developing IoT industry, and paying a fee that is close to or larger than the amount of value being transferred is not logical [5].
- Consensus algorithms like *PoW, PoS* are facing huge setbacks. *PoW* consumes too much energy, while in *PoS*, the rich get richer phenomena appears [32].
- The miners have some power over the network and may be a source of centralization. This means that if a group of miners with a lot of computing power decide to work together, such that they win the majority of the competitions for creating blocks, they can impose their rules on the network. It has been shown that Bitcoin has become more centralized over the years due to the influence of miners with lots of computing power working together [3].
- Blockchain scalability is a concern, and may not be suitable for high frequency transactions as expected in IoT. A block is mined every ten minutes and block size is limited to 1 MB [32]. Also the number of transactions are restricted to 7 per second [32]. The amount of transactions per second in cryptocurrencies using Blockchain is severely limited in comparison to established (online) payment services such as Visa and PayPal [30].
- Although Blockchain prevents various types of malicious attacks, it is not immune to *The 51% attack* [31] [33], *identity fraud* [31], *illegal activities* [31], *selfish mining* [34] and *system hacking* [31].
- Related to privacy, Blockchain is considered to be very safe because users only perform transactions with addresses rather than with real identity. However as stated in [32], Blockchain does not guarantee the transactional privacy since the values of all transactions and balances for each public key are publicly available.

The technical challenges and security issues described above can be a major bottleneck for Blockchain mass adoption. To overcome these challenges, IOTA was proposed.

IOTA is a permissionless distributed ledger that utilizes a novel invention, called a "Tangle", at its core [4]. The Tangle data structure is based on a Directed Acyclic Graph. As such, it has no blocks, no chain and also no miners. This radical architecture change enables IOTA to work quite different from Blockchain. Apart from the data structure, IOTA also achieves consensus differently. As there are no miners, each participant node in the network actively participates in the consensus by approving two past transactions [4]. To pick these transactions, a Tip Selection Algorithm (TSA) is ran. The current proposed TSA is Monte Carlo Markov Chain (MCMC) algorithm. However, until IOTA becomes mature, the *Coordinator* is used for consensus instead of MCMC to prevent certain attacks [8]–[10]. Instead of binary logic, IOTA uses Trinary logic based on trits and trytes. Trinary is mostly used because trinary arithmetic is found to be more efficient than binary and seems more logical [6] [7].

Each node in the graph corresponds to a transaction. A transaction contains a source or destination address and a transaction value. Transactions are bundled in a bundle of inputs and outputs. Inputs are the source addresses and amounts to be transferred from them. Outputs are the destination addresses and the amounts to be transferred to

them. Every edge in the graph corresponds to an approval. A directed edge from node A to node B means that transaction A approved transaction B. A new transaction that has not been approved by any other transactions yet, is called a *tip*. When a transaction is issued and added to the tangle, it approves two of these tips, as shown in Figure 2. When approving, the tips are checked on whether they are valid transactions that do not conflict with the tangle history [5]. Transactions can be issued by nodes in the IOTA network, called *Mainnet* [35].

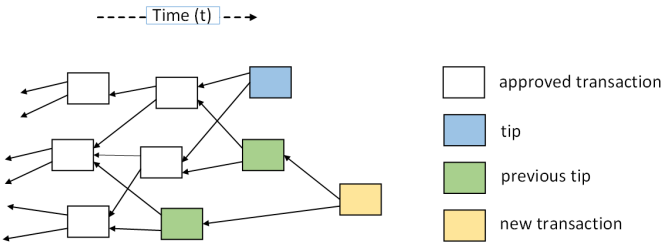

Figure 2. Adding a transaction to the tangle

The unique features that make IOTA superior to Blockchain are [4]:

- Scalability: IOTA can achieve high transaction throughput thanks to its parallelized validation of transactions with no limit on the number of transactions that can be confirmed in a certain interval. This feature of IOTA supports the IoT ecosystem in which large quantities of micro-transactions will be performed.
- Decentralization: IOTA has no miners. Every participant in the network that is making a transaction, actively participates in the consensus. As such, IOTA is more decentralized than Blockchain.
- IOTA has no transaction fees which is more suitable for IoT.
- IOTA utilizes a next generation trinary hash function called Curl-p, which is quantum immune (Winternitz signatures).

III. IOTA Security

The overall IOTA technology sounds as a very groundbreaking concept in the DLT world. Like any other technology, IOTA also runs into technical challenges and security issues [11]–[13]. IOTA is relatively new and still under development, so apart from online blogs, barely any concrete literature is publicly available related to IOTA security [5] [14] [15] [16]. However there have been a few short reports describing possible vulnerabilities in IOTA. Majority of the vulnerabilities identified are concerns rather than practically exploited threats. Based on its lack of tangible evidence or formal proof, IOTA foundation has denied the majority of the security concerns [9] [10] [17] [18]. The aim of this research is to identify IOTA security issues and formally exploit a replay attack.

Some of the alleged IOTA vulnerabilities or technical challenges found in literature and various online sources are: *Curl-P hashing collisions* [11] [14], *Waste money attack* [14], *Steal money attack* [14], *Large weight/outpace attack* [5], *Splitting attack* [5], *34% attack* [8] [21] [22] [23], *Centralization* [14] [16] and *Replay Attacks* [15]. Among all the possible IOTA security vulnerabilities mentioned, we have exploited the replay attack.

A. Replay Attacks

The acceptance and adoption of cryptocurrencies is soaring, so is the continuous stream of attacks. If successfully exploited the consequences can be catastrophic. One of the threats IOTA is facing is replay attack [15]. In the context of IOTA cryptocurrency, replay attack refers to stealing users IOTAs by 'replaying' transaction bundles. Currently, when a bundle is attached to the tangle and it is confirmed, it is possible to attach the same bundle again. If the bundle is still 'valid', it can be confirmed again. Joseph Rebstock introduced this attack and explained a few variants [15]. In his work he stated that when a transaction bundle is sent into the network multiple times it will repeatedly be confirmed, and thus the transactions will be repeatedly executed. This indicates that the replay attack is possible. He explains that IOTA users are vulnerable to the replay attack when they reuse addresses. When one address is reused, meaning that there are funds left on that address after a transaction from that address has been done, all the addresses 'downstream' of that address can be used by an attacker to drain the funds from the vulnerable address. This can be done through so called 'chain replays'. If a vulnerable address does not contain enough funds for a transaction to be replayed, the attacker may decide to top up the address. That will enable the attacker to replay the transaction bundle and steal funds and also get back the funds he used to top up, this makes the cost of the attack equal to zero [15]. Although [15] shows the possibility of replaying transaction bundles, no method was provided for the replay attack. In this paper we propose practical replay attack exploitation methodology and show the feasibility of the attack.

In response to the report about replay attacks, IOTA Foundation indicated that address reuse is not supported by default, and as such IOTA is supposed to be secure. The IOTA Foundation is thus not planning to prevent replay attack for now, because they claim it is not a vulnerability, referring to that addresses are not to be reused [19] [20]. Notwithstanding this, we do consider replay attack as a threat because, **first** the attacker or anybody in general can capture the address reuse transaction and replay it to steal or waste funds from the victim. **Second**, even though address reuse is not recommended it is feasible and thus it may be forced upon victims, which would make the replay attack possible. **Third**, IOTA has re-defined the trust. If it can not secure funds, nobody will be ready to trust it.

IV. IOTA replay attack exploitation

In our proposed methodology, in order to perform replay attack, it is required that the victim reuses the address. In the subsequent sections we will illustrate how the IOTA API uses addresses in transactions bundles and how we modified it such that addresses are reused, and then perform a full-fledged replay attack.

A. Normal IOTA transactions

In normal or default IOTA configurations, to execute transaction(s) the following steps are performed:

1) Using the IOTA API, a user creates a transaction with the amount of IOTAs to be transferred, destination address and optionally a tag.
2) The user can also provide a remainder address which will then be used to send the user's remaining funds to, after the intended transaction. IOTA employs so called seeds to generate an address. So if no remainder address is provided, IOTA uses the seed to generate an address for the user's remaining funds. The input address will be completely emptied.
3) The *prepareTransfer* API function calls *addRemainder* function to empty the input address. In *addRemainder* function, *toSubtract* represents the transfer amount from the input address. *toSubtract* currently equals 0-*thisBalance*, where *thisBalance* is equal to the number of IOTAs currently on the address. *addRemainder* then checks if there will be a remainder based on the intended transfer to destination address and *thisBalance*. If there is a remainder, it creates a transaction to send the remaining funds to the remainder address that was provided or generated as mentioned above. Since this is the function that avoids address reuse, in our methodology we exploit this function to perform replay attack.
4) The *prepareTransfer* API function calls *signInputsAndReturn* function to sign the transactions.
5) The resulting transactions are converted to trytes and bundle is formed and sent into the network.

B. IOTA transaction after forced address reuse

In order to force IOTA API to reuse the input address, we modify *addRemainder* function. Instead of emptying the input address and sending the remaining funds to new address, we modify *addRemainder* as described below:

1) *toSubtract* is initialized to 0, then we check whether the total value to be transferred (*totalTransferValue*) is bigger than current balance (*thisBalance*). If this is true, *toSubtract* is set equal to the balance of the current input address such that this address will be completely emptied. If it is not true, that means that the total value to be transferred is smaller than or equal to the amount of IOTA on the current input address. This means that *toSubtract* can be set equal to *totalTransferValue*, such that only the amount of IOTA necessary for the transfer is deducted. In pseudocode this looks as shown in Figure 3. Modified *toSubtract* now only deduces the intended transfer amount.

Pseudocode: Before updating IOTA API code:

$$toSubtract := 0 - thisBalance$$

Pseudocode: After updating IOTA API code:

$$\begin{aligned} &toSubtract := 0 \\ &if\ totalTransferValue > thisBalance: \\ &\quad toSubtract := 0 - thisBalance \\ &else: \\ &\quad toSubtract := 0 - totalTransferValue \end{aligned}$$

Figure 3. Pseudocode *toSubtract*

2) Input address is reused; we modify *addRemainder* such that no remainder address is used to send the remaining funds to. No transaction for the remaining funds is generated, such that the remaining funds stay on the input address, thus the input address is reused.
3) The *prepareTransfer* API function calls *signInputsAndReturn* function to sign the transactions.
4) The resulting transactions are converted to trytes and bundle is formed and sent into the network.

The difference between the API transaction generating process before and after the API is modified is shown in Figure 4.

C. Demonstration of successful address reuse

In order to practically demonstrate that IOTA indeed accepts reused addresses, we perform the following steps:

1) We have written *sent_single_trans.js* script which is used in combination with updated IOTA API to generate transactions. As updated IOTA API is used, the input address is reused. We test this script on the current IOTA test network to generate a transaction. The IOTA test network is a network of nodes which operate the same node software as in the actual IOTA network. Tangle visualizer *(https://testnet.thetangle.org)* is used to visualize the transaction. The bundle we generated with *sent_single_trans.js* using the default IOTA API is shown in appendix **S.1** and the bundle we generate with *sent_single_trans.js* using the updated API is shown in appendix **S.2**.
2) We have written *store_prepared_bundle.js* script to store the transaction generated in the previous step in the database.
3) We also have written *send_stored_bundle.js* script to read a transaction bundle from the database and then send it into the network. We execute

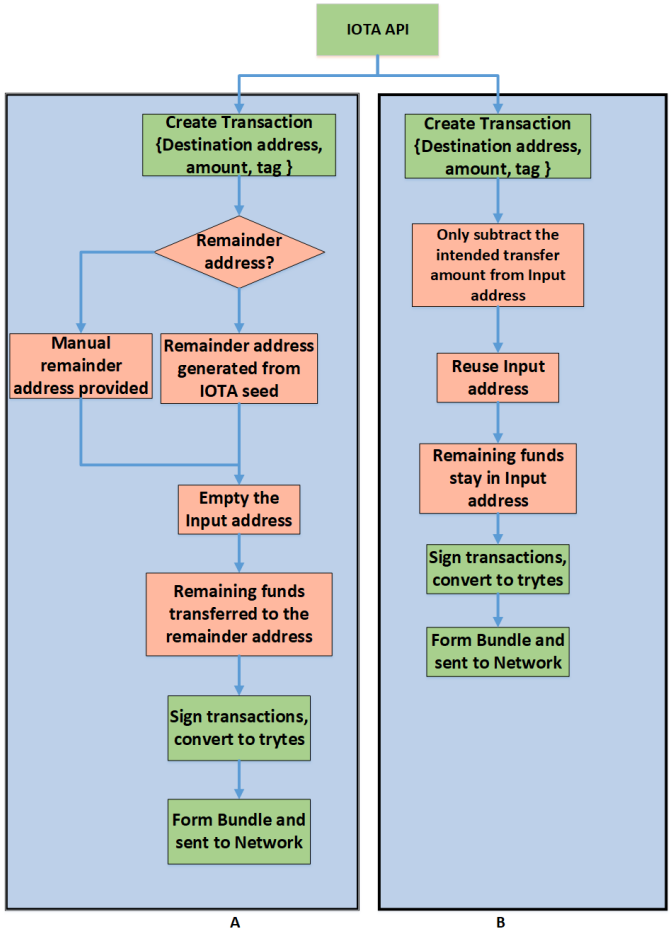

Figure 4. Normal IOTA transaction (A) vs IOTA transaction after IOTA API modifications (B).

send_stored_bundle.js script multiple times to send multiple instances of the same transaction bundle into the network. From the online tangle visualizer, it is obvious that the input address is reused. Also, the same transaction bundle is confirmed multiple times, which confirms that the address reusing transactions are replayed multiple times. Thus IOTA accepts address reuse. We have shown the evidence of successful address reuse in appendix **S.4**, **S.5**. Appendix **S.4** shows transactions from an input address we generated and shows that the input address is not changed and funds are deducted multiple times. Appendix **S.5** shows the transactions received at the destination address and the amount received, also it shows that we replayed the transaction four times and the destination address received the amount four times.

D. Replay attack exploitation

After we have successfully shown that address reuse is possible and transactions are confirmed multiple times, now the other challenges are: *Challenge 1*, how can the attacker identify the address reusing transactions? We easily overcome this challenge by using tangle visualizers like *https://thetangle.org* to find address reusing bundles and retrieve bundle information like *bundleHash*. Transactions that belong to a certain bundle all have the same *bundleHash* property. *Challenge 2*, how can the attacker reconstruct the transaction bundles to perform the replay attack? As mentioned earlier, a transaction is represented by a string of trytes. Essentially, the bundle stored by *store_prepared_bundle.js* and retrieved from the database by *send_stored_bundle.js* differs from the bundle that is finally sent into the network, in the sense that it has a different format and contains different information. An attacker is only able to get the bundle that is finally sent into the network. The methodology we use to overcome this challenge and successfully exploit replay attack vulnerability is the following:

1) Using tangle visualizer, we identify address reusing bundle and retrieve bundle information like *bundleHash*, which allow us to find transactions of a bundle.
2) As IOTA API allows to search for transactions using *bundleHash*, we use the IOTA API to get all the transactions belonging to address reusing bundle. For this step, we have written *get_bundle_and_reconstruct.js*, which gets all the transactions from the tangle that have a certain bundle hash. It then uses these transactions to reconstruct the bundle with the given bundle hash.
3) In order to reconstruct the address reusing bundle, we create a new bundle with similar information as the address reusing bundle for each transaction entry in the bundle like: address, transfer amount, message tag, and timestamp.
4) After adding entries to the new bundle, we add the signatures from the transactions to the corresponding bundle entries.
5) Finally, we convert the bundle entries into trytes. The resulting bundle contains the same transaction trytes for each transaction as the originally found address reusing bundle, and thus it is the same. Therefore the address reusing bundle is successfully reconstructed. Using *get_bundle_and_reconstruct.js* script we can store the reconstructed bundle in database and (or) send it to the network. The reconstructed bundle is send into the network again using IOTA library *sendTrytes* function. The network interprets it as valid bundle and confirms it again.
6) For proof of concept appendix **S.3** shows the address reusing bundle we reconstructed and replayed using *get_bundle_and_reconstruct.js* and shows that it got confirmed multiple times.

V. Discussion

There are various ways an IOTA user may be forced to reuse addresses, possibly without even knowing it. IOTA

wallets or other software that uses the IOTA API, may force a user to reuse address. If a user just provides a destination address and a transfer amount, the software will create the bundle based on that information. The software may then create an address reusing bundle and send that into the network, and in this way force address reuse. There is good probability that this anomalous behaviour of the software would be detected through code analysis. Address reuse will also be visible with the tangle visualizer, which allows the user to check bundles and see their inputs and outputs. If an attacker were to use software to force users to do address reuse, the attacker would first have to make users adopt the software. This will likely not succeed if it is known that the software reuse addresses. So the success of replay attack also relies on the technical knowledge of the user.

A. Variants of the replay attack

Apart from forcing address reuse, there are also other ways to exploit the replay vulnerability:

1) Brute force: To create transaction bundles with the API, a user only needs a seed which has addresses with funds and the user needs to provide a destination address and a value. What an attacker might do is try to create transactions for a guessed seed. It is questionable whether this will be worth it, the probability that a user finds a seed that gives access to funds is likely very low.

2) Past transaction bundles: According to [15], there have been transaction bundles already that reuse an input address. These bundles, if there is still IOTA left on the input address(es), are still vulnerable to the replay attack. Thus, an attacker may search through the tangle and try to find bundles that did address reuse, check the input address(es), and if there are still funds on them the attacker can execute a replay attack.

B. Replay attack prevention

The replay attack vulnerability can be fixed in various ways. In [15], it is suggested to 'keep track of the unique hash of each signed transaction bundle', and then only allow one instance of a bundle hash in a subtangle. To do that bundle hashes would have to be made unique, no collisions should occur, and they should be sufficiently long such that a virtually unlimited number of bundles can be created. Other ways might be to make (signed) transaction hashes unique and allow only an individual instance of each transaction hash in a subtangle. If this is implemented, an IOTA network node should check each incoming transaction to see if heir hash already exists in the tangle. Both ways described above will likely cause quite some overhead.

C. Impact

The replay attack can be considered as the breach of IOTA integrity because the attacker captures the addresses reusing legitimate transactions and performs unauthorized or fraudulent transactions and consequently the victim can lose the funds available on the reused addresses. If the replay attack is performed on large scale, it can have immense adverse impact on IOTA trust.

VI. Conclusion

Literature and a range of online sources have been used to explore potential security vulnerabilities of IOTA. Six potential issues have been identified. The replay attack has been analyzed in depth, and a method to exploit it has been provided, while the remaining five vulnerabilities mentioned still need to be scrutinized. It was demonstrated that the replay attack can indeed be executed. However address reuse prerequisite is severely limiting the attack. It is against the IOTA recommendation to reuse addresses; however, IOTA does not enforce this rule. There are various variants of the attack, but they all rely on addresses being reused. Assuming the attacker does not have access to the victim's software, it will be hard for the attacker to force victims to reuse their addresses. However, the variants of the replay attack attack do not all require to force victims to reuse their address. Therefore, to be more secure, IOTA should find ways to refuse address reusing transactions. It is concluded that IOTA could be made more secure by fixing issues like the replay vulnerability. But based on this research it is also concluded that IOTA is secure, yet only if used and implemented as recommended.

VII. Acknowledgment

This work has been partially supported by the EFRO, OP Oost program in the context of Countdown project.

Appendix
Proof of concept

S.1 With sent_single_trans.js script and using the default IOTA API we generated the following bundle https://testnet.thetangle.org/bundle/RAOYEWNAZYIFEHBUFPYJIL9IQFNLHJILXCOTCABWX9LKDDFYEOXCRBYZPAFCUVBCQXVCSYLWJSZIYJAYA (11-6-2018)

S.2 With sent_single_trans.js script using the updated IOTA API we generated the following bundle https://testnet.thetangle.org/bundle/CKQGQRDNPTEJJBHBJRYL9DAJVBAFZAJLPSQWAPDJLFEATRJCLGBDYQDFYGHOYJBERGSBBLBPDZOOHYFID (11-6-2018)

S.3 Using get_bundle_and_reconstruct.js script we reconstructed the following address reusing bundle https://testnet.thetangle.org/bundle/RJECNEWPNIQJOKTNYEWJBHKFLXNWJFFGDEBYLCHZFAPUGRZONKWKIQSSCTSRQGNBTEHWSIBQCULCFCKAC (12-6-2018)

S.4 Address reusing transactions we generated with our methodology. https://testnet.thetangle.org/address/JUCVGGDXSEEHYSSWAWWUOLGJJXOZZNMOZYKZYMYFNNJBPMPDLJKKJ9WDXLGTCNFFJDCDHIAPNFUADXOSA

S.5 Address reusing transactions we generated with our methodology. https://testnet.thetangle.org/address/HFWAHJEGDJMMMYZAIUTM9UG9NMRPFLGCQQTUDWZBHARBCLDGSUSMZJZPBSSLYYRXOBUMQRSEKVDXTVGDD

References

[1] Natarajan, H., Krause, S.K., Gradstein, H.L. (2017). Distributed Ledger Technology (DLT) and blockchain. FinTech note; no. 1. Washington, D.C.: World Bank Group. Retrieved from: http://documents.worldbank.org/curated/en/177911513714062215 /Distributed-Ledger-Technology-DLT-and-blockchain

[2] Bucko, J., Palová, D., Vejačka, M. (2015). Security and Trust in Cryptocurrencies. Central European Conference in Finance and Economics 2015. Herľany, Slovakia. Retrieved from: https://www.researchgate.net/publication/317955860_Security_and_Trust_in_Cryptocurrencies

[3] Beikverdi, A., Song, J. (2015). Trend of centralization in Bitcoin's distributed network." IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). Takamatsu, Japan. pp. 377-383. doi: 10.1109/SNPD.2015.7176229

[4] What is IOTA. (2018). Retrieved from: https://docs.iota.org/introduction/what-is-iota

[5] Popov, S. (2018). The Tangle. Retrieved from: http://iotatoken.com/IOTA_Whitepaper.pdf

[6] Why does IOTA use ternary based logic? Retrieved from: https://www.iota.org/get-started/faqs (section Advanced)

[7] Hayes, B. (2001). Third Base. American Scientist. 89(6). Retrieved from: http://bit-player.org/wp-content/extras/bph-publications/AmSci-2001-11-Hayes-ternary.pdf

[8] Sønstebø, D. (2017, June 15). The Transparency Compendium. [Blog Post]. Retrieved from: https://blog.iota.org/the-transparency-compendium-26aa5bb8e260

[9] IOTA Foundation. (2018, January 7). Official IOTA Foundation Response to the Digital Currency Initiative at the MIT Media Lab – Part 3/4. [Blog Post]. Retrieved from: https://blog.iota.org/official-iota-foundation-response-to-the-digital-currency-initiative-at-the-mit-media-lab-part-3-6433b55c7d57

[10] IOTA Foundation. (2018, January 7). Official IOTA Foundation Response to the Digital Currency Initiative at the MIT Media Lab – Part 4/4. [Blog Post]. Retrieved from: https://blog.iota.org/official-iota-foundation-response-to-the-digital-currency-initiative-at-the-mit-media-lab-part-4-11fdccc9eb6d

[11] Narula, N. (2017, September 7). Cryptographic vulnerabilities in IOTA. [Blog Post]. Retrieved from: https://medium.com/@neha/cryptographic-vulnerabilities-in-iota-9a6a9ddc4367

[12] Hu, Y.-C., Yip, A., Shin, M. (2017, November 21). IOTA Tangle and Cryptographic Vulnerabilities [YouTube]. Retrieved from: https://www.youtube.com/watch?v=vmwYCjcbUc8

[13] Johnson, N. (2017, September 26). Why I find Iota deeply alarming. [Blog Post]. Retrieved from: https://hackernoon.com/why-i-find-iota-deeply-alarming-934f1908194b

[14] Heilman, E., Narula, N., Dryja, T., Virza, M. (2017). IOTA Vunerability Report: Cryptanalysis of the Curl Hash Function Enabling Practical Signature Forgery Attacks on the IOTA Cryptocurrency. Retrieved from: https://github.com/mit-dci/tangled-curl/blob/master/vuln-iota.md

[15] Rebstock, J. (2018). Replay Attacks in IOTA. Retrieved from: https://github.com/joseph14/iota-transaction-spammer-webapp/blob/master/replay%20attack.md

[16] Wall, E. (2017, June 14). IOTA is centralized. [Blog Post]. Retrieved from https://medium.com/@ercwl/iota-is-centralized-6289246e7b4d

[17] IOTA Foundation. (2018, January 7). Official IOTA Foundation Response to the Digital Currency Initiative at the MIT Media Lab – Part 1/4. [Blog Post]. Retrieved from: https://blog.iota.org/official-iota-foundation-response-to-the-digital-currency-initiative-at-the-mit-media-lab-part-1-72434583a2

[18] IOTA Foundation. (2018, January 7). Official IOTA Foundation Response to the Digital Currency Initiative at the MIT Media Lab – Part 2/4. [Blog Post]. Retrieved from: https://blog.iota.org/official-iota-foundation-response-to-the-digital-currency-initiative-at-the-mit-media-lab-part-2-9ce650ad789c

[19] 'Mix' (pseudonym). (2018, February 21). IOTA Replay Attacks vulnerability. [Blog Post]. Retrieved from: https://thenextweb.com/hardfork/2018/02/21/iota-replay-attacks-vulnerability/

[20] Freiberg,L.(IOTA dev). (2018). Replay Attacks in IOTA. [Reddit response]. Retrieved from: https://www.reddit.com/r/CryptoCurrency/comments/7yw5py/replay_attacks_in_iota_new_vulnerability_report/dujpzk5/

[21] 'Blockknight' (pseudonym). (2017, October 23). What is Iota? Cryptocurrency and The Internet of Things. [Blog Post]. Retrieved from: https://hackernoon.com/what-is-iota-5da4446602a

[22] Scott, J. (2017, April). IOTA Double-Spending Masterclass. [Forum Post]. Retrieved from: https://forum.iota.org/t/iota-double-spending-masterclass/1311

[23] 'Winston' (pseudonym). (2017). XY Attack Vector (IOTA's version of the 34% attack). [Forum Post]. Retrieved from: https://forum.helloiota.com/469/XY-Attack-Vector-IOTAs-version-of-the-34-attack

[24] Blockchain in Logistics and supply chain: Trick or Treat? Niels Hackius , Maritz Petersen Hamburg University of Technology Kühne Logistics University

[25] Harry Machado. Internet of things impacts on supply chain. Retrieved from:http://apicsterragrande.org/images/articles/Machado___Internet_of_Things_impacts_on_Supply_Chain_Shah_Machado_Second_Place_Grad.pdf

[26] A survey of internet of things: Future vision architecture, challenges and services By Dhananjay Singh IEEE member, Gaurav Tripathi, Antonio J. Jara

[27] Dave Evans. The Internet of Things How the Next Evolution of the Internet Is Changing Everything. (April, 2011). Retrieved from: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[28] Calum McClelland. (December 8, 2016). Internet of Things Examples and Applications. Retrieved from: https://www.iotforall.com/internet-of-things-examples-applications/

[29] Blockchain & Distributed Ledger Technology (DLT). (April 12, 2018). Retrieved from: http://www.worldbank.org/en/topic/financialsector/brief/blockchain-dlt

[30] "Raul". Transaction Speeds: How Do Cryptocurrencies Stak Up To Visa or PayPal?. (2018, January 10). Retrieved from: https://howmuch.net/articles/crypto-transaction-speeds-compared

[31] Xu, Jennifer J. 2016 Are blockchains immune to all malicious attacks? https://doi.org/10.1186/s40854-016-0046-5

[32] Zibin Zheng, Shaoan Xie, Hong-Ning Dai 2017. Blockchain challenges and Opportunities: A Survey

[33] I.-C. Lin, T.-C. Liao. (2017) A Survey of Blockchain Security Issues and Challenges.International Journal of Network Security, Vol.19, No.5, PP.653-659, Sept. 2017. doi: 10.6633/IJNS.201709.19(5).01

[34] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, 2017, pp. 557-564. doi: 10.1109/BigDataCongress.2017.85

[35] The IOTA Token. (2018). Retrieved from: https://docs.iota.org/introduction/iota-token/the-iota-token

[36] Blockchain for Lawyers 101: Part I. Dan Tapscott. Retrieved from: http://www.legalalignment.com/blog/blockchain-for-lawyers-101-part-i

[37] Giang-Truong Nguyen and Kyungbaek Kim. (2018) A Survey about Consensus Algorithms Used in Blockchain. Retrieved from: http://jips-k.org/file/down?pn=530

[38] Evers, L., Havinga, P. J. M., Kuper, J., Lijding, M. E. M., & Meratnia, N. (2007). SensorScheme: Supply Chain Management Automation using Wireless Sensor Networks. In Proceedings of the 12th IEEE Conference on Emerging Technologies and Factory Automation, ETFA 2007 (pp. 448-455). Los Alamitos: IEEE. DOI: 10.1109/EFTA.2007.4416802

[39] Dennis JA Bijwaard, Wouter AP van Kleunen, Paul JM Havinga, Leon Kleiboer, Mark JJ Bijl (2011). Industry: Using dynamic WSNs in smart logistics for fruits and pharmacy, In Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems SenSys 2011 (pp. 218-231), Seattle, United States.DOI: 10.1145/2070942.2070965