A Survey on Security Challenges and Solutions in the IOTA

Mauro Conti[a], Gulshan Kumar[a,b], Pranav Nerurkar[c], Rahul Saha[a,b] and Luigi Vigneri[d]

[a]Department of Mathematics, University of Padua, 35131 Padua.
[b]School of Computer Science and Engineering, Lovely Professional University, Punjab, India
[c]Dept. of Data Science, NMIMS University, India
[d]Research Department, IOTA Foundation, Berlin, Germany

ARTICLE INFO

Keywords:
IOTA protocol, Tangle, Security, Ledger, Graph, Coordinator.

ABSTRACT

Wide-scale adoption of the Internet of Everything requires decentralized security, responsibility, and trust among the stakeholders. All these can be achieved by a Distributed Ledger Technology (DLT) backbone. As a mathematical model for enabling this DLT backbone, IOTA's Tangle is gaining popularity due to its scalability and freedom from transaction fees. Unlike blockchain, the Tangle uses a Directed Acyclic Graph (DAG) structure, and its design does not cover essential blockchain pitfalls, including expensive Proof of Work (PoW), limited throughput, high transaction costs, and significant confirmation delays. The original IOTA is evolving into a Coordinator-less environment, the Coordicide. It requires additional modules, such as auto-peering and a reputation system, to fully exploit Tangle's scalability and complete decentralization potential. Nevertheless, each new evolutionary update adds complexity and may introduce security threats. Therefore, the present survey's motivation is a detailed security analysis of the IOTA. To spur developers and researchers' interest and summarize the security status in IOTA, we have drawn the current review. Our survey outlines security vulnerabilities on IOTA and their mitigation strategies and explores several important open directions to be researched further. The vulnerabilities are discussed on both the original IOTA and its upcoming Coordicide version.

In summary, this survey is first in the field for (i) understanding the basic functionalities of the IOTA, (ii) listing the security solutions provided in the literature against the reported and unreported attacks, and (iii) presenting open research questions (RQ) for directing the future investigations on IOTA.

1. Introduction

The success of Bitcoin in the last twelve years has revealed the importance of blockchain technology. A blockchain is a DLT that bundles transactions in blocks and connects them with the existing blocks in a chain upon successful validation through consensus [6, 84]. With the blockchain, it is possible to transparently enable seamless peer-to-peer trading simultaneously, providing security and robustness to the participants. Academia and industry have shown keen interest in extending the applicability of the DLTs to the ecosystems like supply-chain, logistics, and the Internet of Things (IoT) [30, 43]. However, such DLT applications need to maintain globally distributed ledgers in a consistent state.

1.1. Blockchain limitations

In Bitcoin and most of the DLTs, consistency is achieved through PoW. Computing PoW aims to find a number that results in a given number of consecutive trailing zero bits when hashed. DLT participants try to solve the PoW faster than the others to have the right to append new blocks to the blockchain [24] and to obtain the associated reward. The design of PoW in Bitcoin creates the following problems:

- **Limited throughput**: Blockchain's use in cryptocurrencies such as Bitcoin and Ethereum have low transaction throughput [101, 100]. The number of transactions per second (TPS) is 7 for Bitcoin, 15 for Ethereum, 56 for Litecoin. TPS of DLTs is far less than that of Visa's 1600 TPS [5]. The only option for increasing transaction throughput in such DLTs is to increase the block size. However, a linear increase in block size causes a linear increase in each network node's data. Due to an increase in data storage, nodes with limited storage capacity would exit the network [5].
- **Transaction costs**: clients pay a fee for making any transaction on DLTs. High fees lead to a transaction getting accepted and approved faster in the DLTs. Fees act as an incentive for the participants to invest in the infrastructure needed for the PoW. On the other hand, fees create a hurdle for micro payments [61]. Bitcoin's average transaction fee in September 2021 was USD 3.660 [1].
- **Confirmation delay**: addition of a single block at a time leads to delay in confirmation of transactions (1 hour with Bitcoin's six blocks finality rule).
- **Inequity**: peers and miners have separate roles. Furthermore, miners, having high computational power, will take control of the DLT, in practice preventing IoT devices to mine new blocks – hence, leading to centralization.

1.2. From Blockchain towards DAG

IOTA Foundation proposed a protocol named IOTA, a data structure based on a DAG, instead of a blockchain, to

ORCID(s):

[1]https://ycharts.com/indicators/reports/bitcoin_statistics

overcome the four bottlenecks mentioned previously [61, 34, 70] [2]. IOTA uses its cornerstone technology, Tangle, a permissionless, scalable distributed ledger designed for the IoT industry [41, 39, 38]. It is a ledger for storing transactions (sites) issued by IoT devices (nodes). IOTA's consensus mechanism is not based on PoW – hence, neither mining races nor fees are required. Thus, IOTA eliminates the dichotomy of users into transaction issuers and transaction miners. Ideally, each new transaction has to approve two existing tip transactions in the Tangle, creating the typical DAG structure. There are no single chain structures in Tangle, so infinite transactions can attach to the DLT at a time. Each transaction issuer has to invest in PoW to ensure that the DLT accepts the transaction. Hence, PoW is an anti-spamming mechanism and not for achieving consensus. IOTA network's first version was launched in 2015. The current mainnet differs from the original Tangle white paper [61] as it includes a Coordinator (a special module to authenticate transactions), which is currently used to preserve security. However, IOTA Foundation, in June 2021, launched an experimental version of the protocol, called the Coordicide version, which removed the Coordinator. The Coordinator ensured security in the Tangle by creating a consensus on transactions that the IOTA network would accept. However, this temporary module was proposed to be replaced in the Coordicide version in the long run. At the same time, to provide security in a Coordinator-less environment, the Coordicide version introduced a voting-based consensus mechanism (Fast Probabilistic Consensus (FPC), node reputation, auto-discovery of network peers, and additional security modules.

Each update of the IOTA ecosystem affects the network's security, which is undoubtedly the most fundamental aspect of any DLT. Unlike blockchain-based DLTs, security analysis on IOTA is a vast topic that needs to consider generic cyber-attacks that could be launched on the Tangle and unknown attacks possible on the heterogeneous network entities like nodes, transactions, clients, IoT devices, qubic protocol, and oracles. The Tangle white paper [61, 62] provides a limited analysis of how the IOTA features affect the security of the protocol. Even the focus of security analysis in the recent literature is about understanding the implications of IOTA security settings on the growth of Tangle [34] or tackling a specific attack (parasite chain [25, 21]). Although existing research papers have highlighted security vulnerabilities in DLTs in general [24, 23, 10, 87, 9]. There is a need for a survey to analyze the wide range of IOTA security implications, a gap that we aim to fulfill. The distinguishing aspects of the current paper are its narrow focus on providing a state-of-the-art analysis of IOTA and its unique security issues [16].

[2]Inspiration for the name IOTA comes from IoT and the Greek letter iota (which usually refers to an extremely small amount). The name was derived to reflect on IOTA's purpose: connectivity of things through micro and/or zero value transactions. In enabling feeless micro transactions, IOTA enables the IoT. IOTA protocol and IOTA are used interchangeably in the paper

1.3. Our contributions

The key contributions of the present research are:

1. We analyze the current architecture of the IOTA network and its enhanced features for a Coordinator-less approach. Additionally, we enumerate the critical features of the IOTA network and describe the building blocks of IOTA.
2. We describe and analyze both the reported and unreported attacks on IOTA.
3. We also review the solutions for both types of attacks.
4. We identify some open research problems and future directions for the readers and researchers.

For a beginner, the paper provides a starting point to understand the IOTA concepts such as transactions, nodes, PoW, and Tangle. We navigate through the journey of the IOTA ecosystem as it transforms across its different versions (Original IOTA, Chrysalis, and Coordicide). The motivation for different features of IOTA is given, along with the scope for their exploitation and possible countermeasures. This survey findings can facilitate the development of a secure protocol for IOTA applications. A recent work by Y. Li *et al.* [43] also has similar contributions to our work. However, it should be noted that IOTA has a fast-changing landscape, and hence, there is a need for updated surveys to inform about its recent developments.

The remaining paper has five sections. Section 2 describes the building blocks of IOTA and dissects essential components of Original IOTA: transactions and nodes. It also describes the terminologies and jargon associated with the IOTA core protocol. Finally, it has security attacks and countermeasures on the Original IOTA. Features of the Chrysalis version and its issues and strengths are given in Section 3. Security attacks and countermeasures on the Coordicide version are shown in see Section 4. Section 4 also presents an analysis of unreported security risks (unreported till September 2021 as the paper is written in this period). The survey also identifies open RQ, which could be the basis of future research in Section 5. Finally, Section 6 presents the conclusion.

2. Architecture of IOTA 1.0

The purpose of this section is to describe IOTA 1.0 also called Original IOTA and its components: IOTA transactions (Section 2.1), IOTA nodes (Section 2.2), IOTA terminology (Section 2.3), and IOTA attacks and remedies (Section 2.4). Figure 1 shows a high-level overview of the three primary entities initially considered in the IOTA architecture. These are:

1. **Clients**: new users of IOTA system who can send micro-transactions or data transactions to nodes.
2. **Nodes**: connected peer devices who together form the IOTA network and are responsible for enforcing the core protocol.
3. **Tangle**: a distributed ledger which stores immutable transactions and is replicated across the IOTA network.

In particular use-cases, clients and nodes are synonymous, and IOTA considers them as nodes. The goal of IOTA is to enable the Internet of Everything (IoE) devices to transact with each other securely. For instance, in Smart Manufacturing applications, sensors (clients) could store data on the Tangle and sell it to third-party applications. The Tangle ensures that the data stored by sensors is confidential, tamper-proof, and remains available. At the same time, the nodes guarantee trust between the multiple parties by allowing them to use the Tangle as a "single source of truth" without levying any transaction fees.

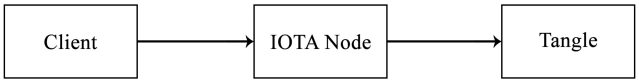

Figure 1: Entities of IOTA

2.1. IOTA transaction

IOTA uses the trinary numeric system, and a single transaction in IOTA is of 2673 tryte-encoded characters. Each tryte consists of three trits. Hence, a tryte can have 27 (3^3) values, i.e., characters A-Z and number 9. The transaction size in the IOTA protocol is equivalent to 1.59 kilobytes. Figure 2 gives the fields of a transaction. On decoding, the transaction trytes form an object with fields shown in Table 1:

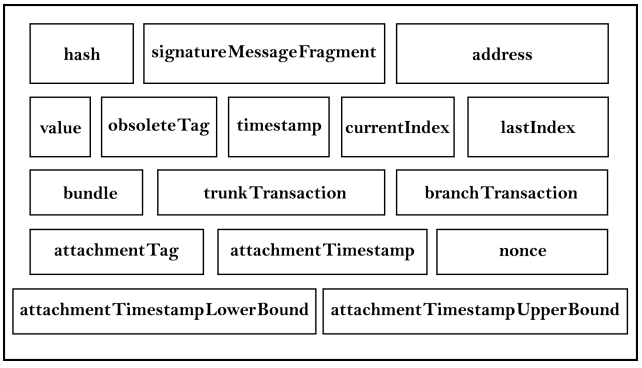

Figure 2: Fields of IOTA transaction

Figures 3, 4 and 5 illustrate the use of cryptography in IOTA. The main applications of public-key cryptography are address generation (see Figure 3), PoW computation (see Figure 4) and signature verification (see Figure 5). For address generation, the client selects a seed and an index which are both converted to trits then combined and hashed into a 243-trit subseed. The subseed is converted to a private key using a sponge function. A private key is split into 81-tryte key fragments. Each key fragment is hashed to generate a key digest which is further hashed and combined to get an 81-tryte address.

In order to attach transactions to the Tangle, clients compute a PoW. For PoW computation, the curl function hashes the transaction to get the last digits of the hash with zeros equal to the Minimum Weighted Magnitude (MWM).

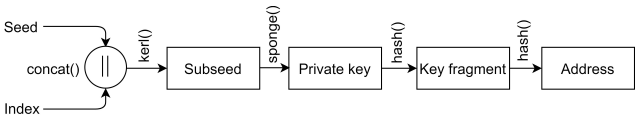

Figure 3: Address generation

If the condition is satisfied, the transaction is attached to the Tangle. The nodes normalize the bundle hash to verify a signature. Then, they select 27, 54, or 81 trytes of the normalized bundle hash depending on the signature's length. These trytes correspond to the number of segments in a signature fragment. The selected trytes of the normalized bundle hash are converted to their decimal values. Each key fragment is hashed once to derive the key digests, which are combined and hashed once to derive an 81-tryte address. If the address matches the one in the transaction, the signature is valid, and the transaction is considered valid.

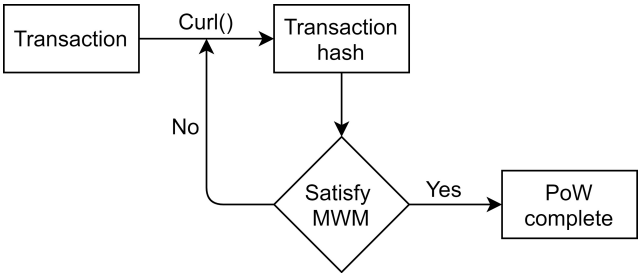

Figure 4: PoW computation

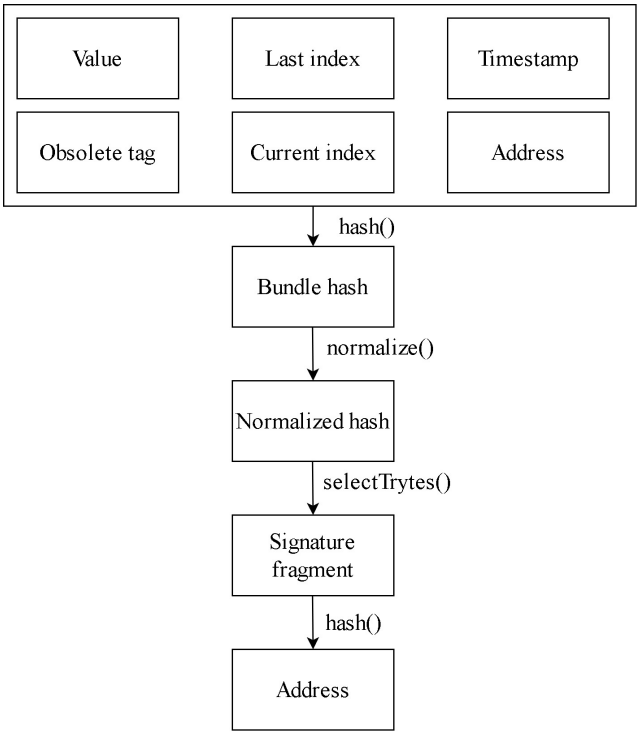

Figure 5: Signature verification

Table 1
Fields of the transaction object

Name of transaction field	Data-type and size (in trytes)	Description
hash	string, 81 trytes	Calculated by hashing transaction fields
signatureMessageFragment	string, 2187 trytes	Contains signature or message. Field is set to all 9's when no message is defined
address	string, 81 trytes	Contains sender address (input transaction) or recipients address (output transaction)
value	integer, 27 trytes	Amount of IOTA tokens to deposit (positive value) or withdraw (negative value)
obsoleteTag	string, 27 trytes	User-defined tag
timestamp	integer, 9 trytes	Unix epoch
currentIndex	integer, 9 trytes	Index of current transaction in bundle
lastIndex	integer, 9 trytes	Index of last transaction in bundle
bundle	string, 81 trytes	Hash of the values of the following transaction fields: address, value, obsoleteTag, currentIndex, lastIndex, and timestamp
trunkTransaction	string, 81 trytes	Transaction hash of either an existing transaction in the Tangle or of the transaction with the next index in the bundle
branchTransaction	string, 81 trytes	Transaction hash of either an existing transaction in the Tangle
attachmentTag	string, 27 trytes	User-defined
attachmentTimestamp	integer, 9 trytes	Milliseconds since Jan 1, 1970 after PoW was done
attachmentTimestampLowerBound	integer, 9 trytes	User-defined
attachmentTimestampUpperBound	integer, 9 trytes	User-defined

2.2. IOTA nodes

Nodes are devices running the node software of IOTA. The software allows nodes to perform operations on the Tangle. Nodes also provide services to clients by exposing APIs. IOTA recommends that each node have a dual-core CPU, 2GB RAM, and SSD storage [65]. Nodes expose two APIs: HTTP API and Events API. The HTTP API allows clients to interact with the Tangle for performing read/write operations. Events API allows clients to poll nodes for new transactions and other events on nodes. Figure 6 shows client interaction with nodes via HTTP API requests and JSON responses. Node API reference version 1.1 offers the following APIs.

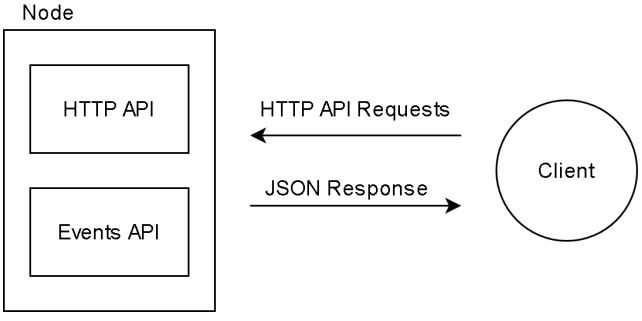

Figure 6: Client interaction with nodes

- `addNeighbors()`: adds a list of temporary neighbors to a node.
- `attachToTangle()`: does proof of work for the given transaction trytes.
- `broadcastTransactions()`: sends transaction trytes to a node.
- `findTransactions()`: finds transactions which contain the given values in their transaction fields.
- `getNodeAPIConfiguration()`: gets a node's API configuration settings.
- `getBalances()`: gets the confirmed balance of an address.
- `getInclusionStates()`: gets the inclusion states of a set of transaction. This endpoint determines if a transaction is confirmed.
- `getMissingTransactions()`: get all transaction hashes that a node is currently requesting from its neighbors.
- `getNodeInfo()`: gets information about a node.
- `getTransactionsToApprove()`: gets two consistent tip transaction hashes to use as branch/trunk transactions.
- `getTrytes()`: gets a transaction's contents in trytes.

Table 2
SI notation for the different units of IOTA token

Unit	Name	Amount of IOTA tokens
Pi	Peta IOTA	1,000,000,000,000,000
Ti	Tera IOTA	1,000,000,000,000
Gi	Giga IOTA	1,000,000,000
Mi	Mega IOTA	1,000,000
Ki	Kilo IOTA	1,000
i	IOTA	1

- `interruptAttachingToTangle()`: aborts the process that's started by the `attachToTangle()`.
- `removeNeighbors()`: temporarily removes a list of neighbors from a node.
- `storeTransactions()`: stores transactions in a node's view of the Tangle.
- `wereAddressesSpentFrom()`: checks if a given address is spent.

2.3. IOTA terminology

Building blocks and related IOTA concepts are Tangle, the procedure for issuing new transactions on Tangle, tips, orphaned transactions, and conflicting transactions, Tip Selection Algorithm (TSA), transaction, height, depth, score, and weights. This section introduces the terminology of the IOTA network in order to familiarize notations used in the IOTA protocol.

2.3.1. IOTA network

The IOTA ecosystem comprises tokens, addresses, a Tangle ledger, wallets, and the Coordinator.

- *IOTA:* the smallest unit of the underlying currency is also called IOTA. The genesis transaction mints all IOTA tokens; therefore, every IOTA token can be traced back to genesis block [5].
- *Units of IOTA tokens:* table 2 gives the System International (SI) prefixes for the units of IOTA token. 2.7Pi is the maximum supply of IOTA tokens in a Tangle. Crypto-exchanges sell in Mi.
- *Address:* an address is generated from a public key and is associated with IOTA tokens which are needed for sending and receiving IOTA tokens. It is a string of 81 trytes.
- *Address generation:* IOTA users memorize a single seed (random string), and addresses needed for a transaction can be generated from that seed. Wallets must support the automatic handling of addresses and discourage address reuse. A seed selects an index between 0 and 9, 007, 199, 254, 740, 991, and a security level between 1 and 3 to generate an address. The same index and security level generate the same address.
- *Private key:* it is generated from a random seed phrase (master key) and can generate an address. A single private key can generate a single address. It proves ownership of an address or message. A seed is 81 trytes. Total seeds possible are $8.7x10^{115}$. A wallet is an application that manages users' seeds, private keys, and addresses. A single seed can generate 9^{57} private keys.
- *Spent address:* address used in an input transaction.
- *Faucet:* it transfers IOTA tokens to an address.
- *Wallet:* these store information about IOTA addresses and balances. They store the seed phrase and keep track of the clients' accounts. There are two types of wallets: Hot wallet and Cold wallet. A hot wallet is connected to the internet and keeps track of user balance. A cold wallet is not connected to the internet and storing account data and signing transactions offline.
- *Wallet types:* four modes for accessing wallets are: Command-line interface (CLI), Graphical user interface (GUI), Paper-mode, and Hardware mode.
- *Tangle:* as shown in Figure 7, there is a "genesis" site (empty block) which is approved by transactions 'A', 'B' and 'C'. Directed edges show approvals, and indirect approvals are indicated by a directed path of length > 1, i.e., from D and E to genesis transaction. To issue transactions, nodes have to approve two other transactions; thus, contributing to the network's security [25]. The two transactions need not be distinct; Hence, 'A' has approved genesis block twice, whereas 'B,' 'C,' 'D' and 'E' have approved two separate transactions.
- *Private Tangle:* It is a permissioned DLT controlled by an entity that restricts participation to available nodes.
- *Tips:* transactions which are not approved yet by other transactions. Figure 7 shows the blocks D and E, which are the tips.

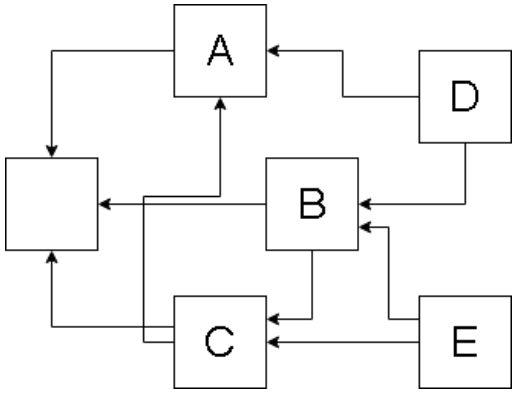

Figure 7: Tangle - DAG

- *Coordinator:* it is a trusted node and regularly adds special transactions to the Tangle. Such transactions are also called milestones and are irreversible. Thus, the Coordinator increases the security of Tangle. The Coordinator is a temporary measure to guarantee security. IOTA Foundation is working to remove this component with the project Coordicide deployed for testing on the GoShimmer node software testnet [69].
- *Subtangle:* section of the Tangle that contains transactions between milestone and tip. Transactions added by the Coordinator confirm an entire sub-tangle.

2.3.2. Transactions

Transactions are for storing sensor data and encrypted messages on the tangle or withdrawing and depositing IOTA tokens. The core IOTA protocol performs several checks described below to ensure the integrity of transactions.

Types of transactions: a bundle in IOTA consists of multiple transactions. Each bundle is atomic, i.e., all transactions in it are accepted, or none are. Types of transactions are:

- Output transaction: IOTA tokens are transferred.
- Input transaction with a positive value: IOTA tokens are transferred to a new address controlled by the sender.
- Input transaction with a negative value: transactions completely spending the account's balance.
- Zero-valued transactions: these transactions have a value of 0 and make use of the *signatureMessageFragment* to store data on the Tangle.

Masked Authentication Messaging (MAM): these are transactions that contain messages instead of value transfer.

Transaction approval time: transaction approval time ranges between $\theta(\lambda^{-1})$ to $\theta(h)$ where, λ rate of incoming transactions and h is average computation/propagation time for a node [69]. The current transaction approval time for IOTA is 10 seconds after the launch of Chrysalis (the intermediate stage before Coordicide is complete) and the TPS during testing touched 1500. Before August 2020, the TPS was 20, and the transaction approval time was 80 seconds.

Procedure for issuing new transactions on Tangle: ideally, to issue new transactions, a node must choose two distinct [58] transactions to approve, it must check if the two transactions are conflicting, and then to issue a transaction, nodes must do a PoW (hashcash) [8]. The validating node must check all previously made transactions of the sender address to verify a transaction.

Orphaned transactions and conflicting transactions: conflicting transactions are double-spending transactions. If A and B are conflicting transactions, then Tangle participants use a rule to decide if A will be approved or B. The rule states that a tip will run a TSA multiple times. If A is selected more than B in TSA runs, A will be approved, and B will be orphaned. Orphaned transactions will not be directly or indirectly approved by tips henceforth [28].

Protocol for transferring tokens: a transaction from Alice's wallet to Bobs is a transaction bundle of three transactions. One for sending IOTA tokens to Bob, the second one for spending all the remaining IOTAs in Alice's wallet, and a third one which is a meta transaction and holds the second part of the signature. Thus, the actual size of a basic Alice-to-Bob transaction results in 8019 trytes (4.77 kBytes). The memory size of a transaction creates the possibility for memory exhaustion attacks targeting nodes.

2.3.3. Nodes

Nodes are IoT devices that issue transactions or propagate new transactions. Nodes must remain active by propagating transactions; otherwise, neighbors may drop them from the Tangle network. The decision to drop a node is based on the number of new transactions received from it. Each node calculates this statistic used to decide when to drop a "lazy" neighbor.

Node software: nodes are the core of the IOTA. They have read and write access to Tangle. Their responsibilities are: (i) attaching new transactions to the Tangle, (ii) synchronizing with the rest of the network, (iii) deciding which transactions are confirmed, (iv) keeping a record of the balances on addresses, and (v) exposing APIs for clients. All nodes run one of the recommended software: (i) Hornet, (ii) Chronicle, (iii) GoShimmer, and (iv) IOTA reference implementation (IRI). Hornet is the most up-to-date, Chronicle is an addition to Hornet, which allows storing transactions in a distributed database, GoShimmer is Coordinator-less implementation under testing, and IRI is deprecated.

Permanodes, common nodes, and local snapshots: a common node only stores transactions back to the last milestone set by the Coordinator, or with the introduction of IRI 1.6.0, a node can manually adjust the period for making local snapshots. Whenever a milestone is reached, only the account balances are stored to minimize the necessary storage requirements for a common node. So to access data from a previous checkpoint, a common node relies on permanodes. Permanodes are computers with extensive storage and bandwidth capacity that store the entire history of the Tangle. An alternative is to subscribe to a specific tag. Whenever an incoming transaction features this tag, the node stores it locally. This operation requires additional hardware with a higher storage capacity that monitors the Tangle and stores the desired data.

Entry nodes: these nodes also run the node software; however, they do not participate in gossiping or consensus. Their role is to send a list of neighbors to IOTA nodes to connect. Entry nodes are regulated and monitored by the IOTA Foundation.

2.3.4. Tangle

These mechanisms ensure the integrity of transactions in the IOTA network.

MWM: a setting for PoW which is checked for deciding whether transactions should be attached to tangle. It is the number of trailing zeros that a transaction hash must have. To calculate the PoW, a node converts all the transaction fields' values to trits and hashes using the Curl function. This process continues until the transaction hash ends in the same number of 0 trits as the MWM. So, MWM is proportional to the difficulty in computing PoW and is a vital instrument to prevent spam in the IOTA network.

Confirmation confidence: in the original Tangle white paper [61], confirmation confidence was proposed as a setting used to decide the trustworthiness of a transaction. Higher the confirmation confidence for a transaction lesser the chance for it to be orphaned. For a transaction X, the below procedure was to be used to calculates it:

1. The TSA is run 100 times.
2. The number of tips that approve transaction X is counted.
3. Every tip is weighted by the likelihood that it will be accepted in the future.
4. The confirmation confidence of transaction X is the fraction of approving transactions.

Confirmation confidence was a simple measure that would have been suitable for earlier days of IOTA when TPS was 5-20. However, this setting was not implemented as it was time-consuming. IOTA network has two settings to decide the trustworthiness of a transaction - Oldest transaction root snapshot index (OTRSI) and the Youngest transaction root snapshot index (YTRSI). In the past cone of a transaction, the index of the milestones is checked. OTRSI is the lowest milestones index, and YTRSI is the highest. As milestones indices are synonymous with timestamps, the lower the index older the milestone will be. A trustworthy transaction is one with low OTRSI and low YTRSI. Lazy transactions would have high YTRSI. Semi-lazy transactions will have a high OTRSI. Ideally, OTRSI and YTRSI should be low to be a non-lazy transaction. Due to the distributed nature of the IOTA network, the core protocol cannot bind nodes to have fixed OTRSI or YTRSI settings. Attackers may use this loophole to their advantage. These settings bring forth the importance of the role of a Coordinator, and alternatives will be needed in a Coordinator-less protocol.

Weight and cumulative weight: weight of a transaction is the amount of work an issuing node invests in it. Weight can be in values of 3^n. Cumulative weight is the transaction weight plus the sum of weights of transactions that approve that transaction directly and indirectly. Figure 8 shows weight (top left) and cumulative weight (bottom right) of transaction B as 1. After D and E approve B, the cumulative weight of B changes to 3.

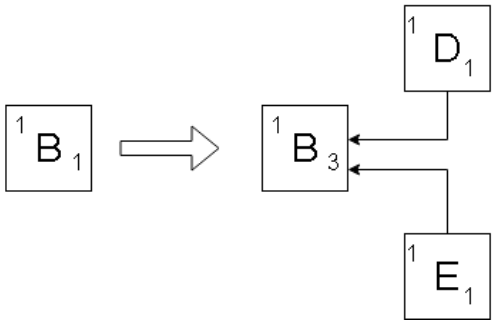

Figure 8: Weight and cumulative weight

Height, Depth and Score: height is the length of the longest path to the genesis. Depth is the length of the longest path to a tip. The score is the sum of all weights of transactions approved by a transaction plus its weight. In Figure 7, height of A is 1 and depth is 2 ($E \rightarrow C \rightarrow A$). From Figure 8, score of D is 2. Linked with height and depth is the concept of the "past cone" of a transaction. It is a set of transactions directly or indirectly referenced by a transaction in the Tangle (including itself). For instance, in Figure 7 the past cone of D is A, B, C, D. In the IOTA core protocol, the past cone is considered for confirmation.

TSA: TSA is used to decide the tips or transactions to which a new transaction will attach itself. The IOTA core protocol does not enforce TSA so that individual nodes may decide TSA. However, IOTA suggests Monte Carlo Markov Chain (MCMC) algorithm. MCMC places random walkers at certain sites on the Tangle. These walkers navigate towards the tips selected for attachment for a new transaction. Random walks can be biased towards transactions with higher cumulative weights. An α parameter controls this bias. When $\alpha = 0$, the random walk is unbiased, and the random walk is biased for $\alpha > 0$. Another TSA strategy is Uniform Random Tip Selection (URTS) which selects tips uniformly at random from a set of available tips [40]. The TSA in production till August 2020 in IOTA cryptocurrency was MCMC with $\alpha = 0.001$ [2]. A TSA is executed each time a node issues a transaction. With the launch of the Chrysalis, the current TSA is a variant of URTS.

2.3.5. Smart contracts and Qubic protocol

IOTA network also aims to introduce functionality such as Smart contracts and Qubic protocol to allow parties to exchange value without needing a trusted intermediary.

Smart contracts: execute transactions on Tangle between two parties without the need for third-party.

Qubic protocol: quorum-based distributed computing protocol for handling smart contracts on IOTA. The following example helps in understanding the building blocks of QBC protocol:

- a logistic company and a consumer establish a "smart contract" for transporting goods to the destination using route *A*. The cost will be calculated based on traffic congestion on route *A*.
- autonomous cars are "oracles" that decide traffic congestion on route *A*. If the "quorum" is set to 2/3 and 2/3 of the cars on route *A* respond to a high degree of traffic, the Tangle registers this information.
- a "qubic" is issued for analyzing traffic congestion and is outsourced to an "assembly" that can efficiently compute this task.

Oracles: real-world entities which collectively form an "assembly". Each oracle is asked to collect an object's data and post the result (qubic) on Tangle. After all members in an assembly post their results, a decision is made on accepting the results. Based on how many oracles in assembly post the same result, acceptance is decided.

Quorum: minimum number of oracles in an assembly that should agree on a result to get it accepted.

Qubic: request for data of an object from the oracles. A node that issues the request (qubic) is called qubic owner. For every qubic, a reward is defined. This reward is split among all oracles that post the result. This promotes honest behavior, as an oracle is not rewarded when posting a faulty result.

ZNET: a zero-valued testnet of IOTA for prototyping and educational purposes.

2.4. Security issues and solutions for Original IOTA

Further, we describe the attacks against the Original IOTA and possible counter-measures available in the literature. We also describe the feasibility and impact of those security problems. Vulnerabilities exist in the IOTA network, which provides scope for severe or mild attacks. Severe attacks that would need protocol safeguards have occurred on the IOTA network and were reported. Additionally, attacks due to human factors such as end-users carelessness also need severe consideration.

2.4.1. Attack vectors for cyber-attack on tangle

The three basic strategies or attack vectors for cyber-attack on Tangle that the attacker may employ to disrupt the IOTA network or any private tangle are:

Attack vector #1 - Eclipsing: aim is to isolate and attack a specific user rather than the whole network.

Attack vector #2 - Byzantine node creation: a node which does not forward messages to other IOTA participants or sends conflicting messages to different IOTA participants.

Attack vector #3 - Sybil identities: an attempt to gain control over a peer-to-peer network by forging multiple fake identities.

Multiple attacks are possible on the IOTA network by using eclipsing, byzantine nodes, or Sybil identities. Prominent amongst these attacks is the *Double spending attack*. In this attack, the adversary may transfer the same IOTA tokens to different users. Steps to perform a double-spending attack are given below:

Step 1: Attacker sends a payment to the receiver, and the receiver sends the goods after confirming the cumulative weight of the transaction is high.

Step 2: Attacker issues a double-spending transaction and many empty transactions to approve the double-spending transaction. The attacker may also use considerable computing power to issue a double-spending transaction. Such a transaction will have a considerable weight.

A fraudulent transaction may grow, and the legitimate one gets orphaned. The time by which such an attack will succeed is $3^{\frac{\lambda w}{\mu}}$. μ is the computing power of the attacker, λ is the arrival rate of transactions, and w is the mean weight of a generic transaction [34].

A second way by which double spending is possible is through a *Splitting attack*. In a high load regime, a splitting attack is possible in which an attacker may try to split the Tangle into two branches. The attacker will place at least two conflicting transactions at the beginning of the split to prevent an honest node from effectively joining the branches by referencing them simultaneously. Then, the attacker hopes that roughly half of the network would contribute to each branch to be able to "compensate" for random fluctuations, even with a relatively small amount of personal computing power. If this technique works, the attacker will spend the same funds on the two transactions. A third attack is the *Large weight attack*. This is a strategy in which an attacker can invalidate a transaction with high confirmation confidence. The steps for the attack are given below:

Step 1: The malicious user waits until a transaction has a high enough confirmation confidence.

Step 2: The attacker uses its computational power and issues a conflicting transaction with a considerable weight followed by many more transactions. This transaction does not approve the first transaction, and thus, they compete with each other for finality.

Securing Tangle against double-spending, splitting, and large weight attacks could be achieved by multiple means. Currently, the Coordinator acts as the first line of defense

against double-spending, splitting, and quantum computations. The Coordinator sends signed transactions called milestones that nodes trust and use to confirm transactions. Transactions in the Tangle are considered for confirmation only when a milestone, validated by nodes, references them directly or indirectly. To allow them to recognize milestones, all nodes in the same IOTA network are configured with the Merkle root address of a Coordinator that they trust to confirm transactions. Using this address, nodes can validate the signatures in milestones to verify whether their trusted Coordinator signs them. To ensure that new transactions always have a chance of being confirmed, the Coordinator sends indexed milestones regularly. This way, nodes can compare their milestones' indices to check whether they are synchronized with the rest of the network.

Parasite chain attack is also possible with the use of Sybil identities. The attacker uses this strategy to improve their transactions' confirmation rates, which may not be confirmed by the IOTA network legitimately. The steps to achieve a parasitic chain attack are given below:

Step 1: An attacker builds a subtangle that references the main Tangle to gain a higher score.

Step 2: Attacker also issues many artificial transactions which approve older transactions made by him.

Step 3: As the parasite subtangle grows, the attacker hopes to orphan the main Tangle [80].

Figure 9 illustrates a parasite structure (shown within the oval) broadcast by the adversary shortly before the next milestone. The parasite structure is attached to a milestone (shown by "*") and does not confirm any honest transaction. Such a structure aims to maximize the probability of selecting a tip in the parasite chain.

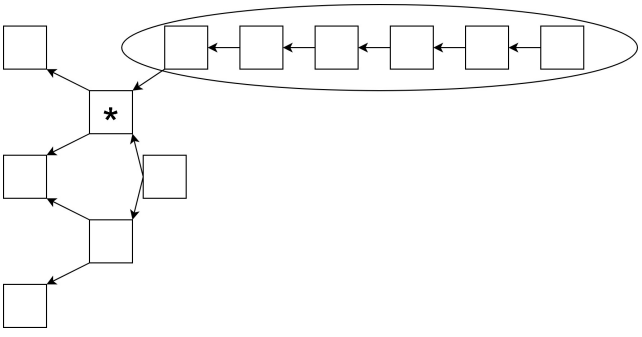

Figure 9: Parasite chain

Using parasite chains, another security vulnerability named "Feather attack" is possible. In such a case, an attacker may attempt to approve their double-spending if less computation power is at his/her disposal. The steps to launching a feather attack are given below:

Step 1: Two double-spending transactions A, B are sent by the attacker.

Step 2: Attacker waits till A has a high cumulative weight then builds a parasite Tangle around B.

Step 3: A parasite chain is built on B such each site confirms the previous one and a site confirmed by A.

Step 4: Attackers hope that with such a parasite chain, A will be orphaned, and the double spend will be approved.

V. Attias *et al.* [2] demonstrate that if MCMC is used as TSA, then the attacker has to generate a parasite Tangle with several sites less than the number of sites confirming A. Thus, an attacker needs less computation power to complete a double-spending attack successfully.

A. Cullen *et al.* [25] proposed a First-order MCMC TSA for securing Tangle against parasite chain attack. The current MCMC algorithm used for tip selection favors transactions attached to heavy subtangles, i.e., cumulative weights are high. The cumulative weight of a transaction in a parasite chain grows linearly with a rate equal to the computing power of the attacker, μ. While the main Tangle will grow at the rate of the hashing power of the rest of the network, λ. The First Order MCMC prioritizes the transactions where cumulative weight grows λ, so the parasite chain is penalized.

2.4.2. Failure or Voluntary halting of the Coordinator

Trinity wallet for IOTA was attacked in February 2020 [66] via a dependency from integrating a third-party service. The attacker transferred 8.55 Ti in IOTA tokens from vulnerable trinity wallets. During the trinity attack on IOTA, the Coordinator was halted by the IOTA Foundation to prevent the attacker from spending stolen IOTA tokens. Consequentially, without the Coordinator, even the genuine clients could not transact, and the entire IOTA network was stopped. Therefore, a Coordinator introduces a central point of failure for the IOTA network. To prevent such a central point of failure from disrupting the entire network, organizations deploying their private tangles would require precautions to safeguard Coordinators from failure or sabotage.

Securing the IOTA network is the responsibility of the Coordinator. This scheme creates an excessive dependence on the Coordinator. Alternative solutions introduce multiple Coordinators who have the effect of reducing centralization. "Stars" is one such proposal by IOTA Foundation. Certain participants in the Tangle will be given equivalent status as the Coordinator. These participants can be well-known or reputed entities such as government agencies, regulators, or legal entities. Tangle shall notify them as "Stars," giving them authority to issue milestones. A benefit of this approach is the reduction of dependence on the Coordinator. Also, as both "Stars" and the Coordinator can create milestones, more milestones would be issued, increasing the network's confirmation rate.

2.4.3. Lazy tips or greedy attachment attack

An attacker may use the lazy tips or greedy attachment attack strategy to get their transactions approved faster. Lazy tips are issued by nodes that approve only a fixed pair of old transactions already approved by others. Such nodes do not contribute to the security of the Tangle. S. Popov *et al.* [70] demonstrate that approval time for transactions made by nodes following a selfish strategy may be less than that of honest nodes by up to 25%.

As IOTA does not enforce any tip selection strategy, rouge nodes can perform greedy attachment attacks. Tangle recommends an MCMC algorithm for tip selection to counter such an attack. It is demonstrated that if the majority of the nodes follow the MCMC strategy in an IOTA network, the greedy attachment attack will be mitigated.

2.4.4. Conflicting transactions

An attacker may use a conflicting transactions strategy to spend the same IOTA tokens in different transactions. Steps needed to carry out a conflicting transaction attack on the network are given below:

Step 1: Two conflicting transactions are created by a dishonest node and sent to two different honest nodes.

Step 2: If network latency is high, then half the honest nodes will receive the first transaction while another half will receive the second.

Step 3: When a node receives both conflicting transactions, it will be assumed that one of the transactions amongst them is correct.

Step 4: Meanwhile, all sites that confirm the second transaction will be discarded, causing loss of hashing power and increase in confirmation time for the discarded transactions as reattachment must occur [16].

Mitigating conflicting transactions is possible by relying on the Coordinator and milestones. Transactions can be confirmed only if they are referenced directly or indirectly by milestones. If there is a double spend in the referenced transactions, then nodes and Coordinator form a consensus that the first transaction that tries to transfer IOTA tokens should be confirmed, and the remaining transactions transferring the same IOTA tokens should be orphaned.

2.4.5. Blowball attack

A blowball is a subtangle formed when many tips refer to a single transaction. Usually, an attacker would launch several "lazy" transactions that only attach to a milestone. Figure 10 illustrates the blowball attack in the tangle. Here the adversary broadcasts spam transactions (shown within the oval shape) that attach to the milestone (shown with "*"). These transactions do not contribute to the IOTA network's security and lower the confirmation rate of the network [91].

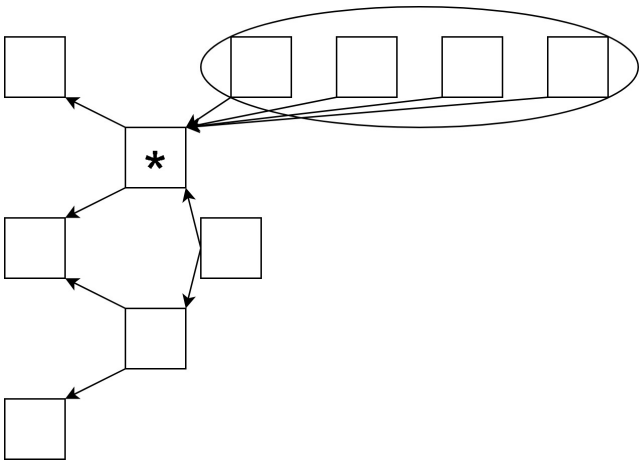

Figure 10: Blowball attack

The blowball attack's goal is to lower the overall confirmation rate of the IOTA network. Hence, a solution should aim to increase the confirmation rate. This means having a milestone that confirms a pair of transactions referencing (directly or indirectly) the highest number of transactions in the network. The score of the transaction could be helpful to implement this strategy. The score could be used as a heuristic to identify transactions connected to many transactions. Thus, the transactions with the highest scores would be potential tips to attach a milestone and boost the overall confirmation rate of the network.

2.4.6. Attacks on nodes

IOTA does not enforce security requirements on nodes. Nodes are ordinary devices and are vulnerable to attacks. Attacks are possible if nodes leave unused ports open. Secondly, to log into nodes, SSH is used. This may allow the attacker to access nodes by impersonating the user to whom the SSH key belonged. Denial of service (DoS) and distributed denial of service (DDoS) attacks are performed on nodes to force a disruption in the IOTA network.

Clients' security against node vulnerability can be ensured by following certain security practices. Attackers can abuse any open ports on the nodes. It is essential to secure the nodes against attacks on unused open ports. So all ports except those which are in use should be closed. A firewall is the first line of defense against port attacks. All operating systems include firewall options. By having a firewall in place, it is possible to completely block unused and unnecessary ports.

2.4.7. API attacks

An organization is setting up a node and making it public to the Internet will face severe risks because it allows anyone with the node's IP address or domain name to spam API requests to the node. Some API endpoints such as `attachToTangle()` are resource-intensive, and too many of these requests can take a node offline.

Security of the API on nodes can be improved if we observe the following practices to secure the API: (i) Whitelist IP address, (ii) Implement basic authentication, (iii) Limit API requests from each user. The node software does not enable basic authentication, whitelisting IP, and limiting API requests by default. Whitelisting IPs give specific IP addresses access to the node. Basic authentication is helpful so that any user without credentials cannot call APIs. Limiting API requests from a user ensures safety against DoS attacks. A reverse proxy server can be used for protection against such attacks.

2.4.8. MQTT broker vulnerability attack

Nodes may allow users to subscribe to events using the MQTT protocol. The default setting of the MQTT broker is to send data using the unsecured channel. This vulnerability could affect confidentiality.

A solution for securing the MQTT broker could be modifying the core IOTA protocol to use a secure TLS channel. The secure channel could be used to send clients updates to protect the confidentiality of data sent by the MQTT broker instead of an unencrypted channel.

2.4.9. Bootstrapping attack

An attack where a node downloads malicious snapshot files, including invalid transactions and balances. A solution against the Bootstrapping attack would be to retrieve the requested data from several permanodes simultaneously, followed by a comparison of the responses received [5].

2.4.10. Reusing address attack

IOTA uses the Winternitz one-time signature scheme (W-OTS) to generate digital signatures. This signature scheme is argued to be quantum robust. However, the scheme also reveals an unknown amount of the private key. So, a brute force attack can reveal the private key. As a result, theft may occur if the client uses the same signature for receiving IOTA tokens in multiple transactions.

Currently, IOTA follows the account balance model where each address is associated with a single value that is its current balance. Therefore, a ledger state is a dictionary of addresses and their corresponding balances, i.e., address_1 $\Longrightarrow$ balance_1, address_2 $\Longrightarrow$ balance_2 and so on. In DLTs with an account balance model, there could be an attack related to reattachments. If somebody ever receives funds on an address that has already been spent, anybody can reattach the previous spend and empty the address again (even without having access to the address's private key). This has already been used as an "attack vector" when users did not follow the advice to use addresses only once (i.e., people receiving donations or other payments after the address has been used) [49].

Although the critical strategy to prevent such attacks could be end-user education. However, the IOTA foundation could bring official IOTA wallets must securing wallets from reusing address attack IOTA wallets. The solution could be to create an abstract layer that generates a new address each time to send and receive tokens, thus strictly prevent against address reuse. A second method to prevent these attacks could be the adoption of the unspent transaction output (UTXO) model (see Section 3.1).

2.4.11. Social engineering attacks

IOTA Foundation operates a Discord server for information exchanges amongst its community. In such an insecure environment, attackers may find victims by asking people to share their IOTA seed or ledger recovery phrase. Attackers may also try to pretend to IOTA Foundation members and mislead users in the pretext of providing support to transfer or exchange funds. Imposters may also launch websites or external services to generate seeds for users or create Trojan IOTA wallet software.

Securing clients against social engineering is a mammoth task and requires concerted efforts from the IOTA community. To keep users and their IOTA coins and other crypto-coins safe, IOTA Foundation should advise new users never to give anyone IOTA seed or ledger recovery phrases. Not even if a person pretends to be from the IOTA Foundation. Suitable warnings should be provided for new users advising them not to trust anyone for advice on transferring or exchanging funds. Users should be warned against websites or external services to generate seeds and recommend downloading wallet software only from the IOTA Foundation's sources.

2.4.12. Lazy tip attack

Currently, in Chrysalis, nodes categorize tip transactions into three groups to better the confirmation rate. These groups are:

1. **Non-lazy**: tips whose both parents have been confirmed by a recent milestone
2. **Semi-lazy**: tips that have one parent confirmed by a recent milestone
3. **Lazy**: transactions whose an old milestone has confirmed both parents

Due to such categorization of transactions, the TSA in Chrysalis has multiple complications [76] such as computationally expensive computation for OTSRI and YTSRI, difficulty in promotions of transactions from categories such as lazy or semi-lazy to non-lazy because an old transaction will have a negligible chance of being approved. Finally, the distinction, lazy and semi-lazy, created by TSA in transactions is a feature with little relevance.

Assume a rogue node that deliberately attaches transactions of genuine users to subtangles confirmed by old milestones. This will reduce the confirmation rate of the network as such transactions will be classified by other IOTA nodes as "lazy." Reattaching such transactions may cause latency in the network, and also such transactions would not contribute to the security of the network. Another possible attack in the network is if a rogue node floods the network with semi-lazy tips. The TSA may select such tips for attaching new transactions. However, with the arrival of new milestones, such semi-lazy tips may become "lazy," orphaning the transactions attached to them.

The task of mitigating lazy tip attacks is challenging to implement, requiring new modules to the IOTA software. A new setting named Youngest Referenced Milestone (YRM) could be defined for a transaction that is the index of the most recent milestone in the past cone. For example, a transaction directly referencing Milestones 10 and 8 will have YRM = 10. A transaction referencing this and an older transaction will also have YRM = 10. The TSA then uniformly picks a tip satisfying the two constraints, i.e., the YRM of a tip should be close to the current milestone index at the time of arrival, and the tip must not be attached below a particular depth.

The advantages of such a solution to reduce the lazy tip attack are that the YRM can be recursively calculated and does not change with incoming milestones. Furthermore, The tip list can be recursively maintained, and it does not need to be updated when a milestone arrives. The difficulty in promoting transactions is resolved. In the solution, a transaction is only eligible for TSA if it has at least one parent attached to a subtangle confirmed by the most recent milestone and is thus helping the network.

However, the main drawbacks of the solution are additional computations to create a method for identifying transactions below a maximum depth. Thus, there is a need for a method with low computational cost to compute which tips are below max depth. The solution can create another attack where a lazy node could always select the last milestone for its two tips [76] not contributing to the network's security.

2.4.13. Conflict spamming attack

An adversary broadcasts two (or more) conflicting, yet individually valid, transactions (C_1, C_2 in Figure 11) simultaneously. Although both C_1, C_2 consume the same tokens, only one of them should be accepted and the other orphaned. In practice, both these transactions may be selected by honest nodes performing tip selection, leading to a situation in which each of these conflicts will have multiple transactions approving it (marked by blue and orange circles in Figure 11). This introduces a Tangle split since no valid transaction can ever approve more than one of these conflicts. Thus, eventually, every transaction approving any but one conflict will become invalid and require reattaching. Even two conflicts alone can potentially "invalidate" hundreds of honest transactions [90]. Such an attack vector is possible for any consensus mechanism on the Tangle in which conflicts remain unresolved for a certain amount of time. The Original IOTA was vulnerable to this attack vector.

The Hetfield solution for conflict spamming is a potential solution based on the principle "Ignore everything conflicting." Each transaction gets an additional flag, whether it is locked or not. Then, the tip selection algorithm never selects a tip that directly or indirectly approves a locked transaction. As potentially invalid transactions are locked, this reduces the chance of reattachments [94]. When two disputing transactions are received quickly, both will be ignored as their flags will be set to lock.

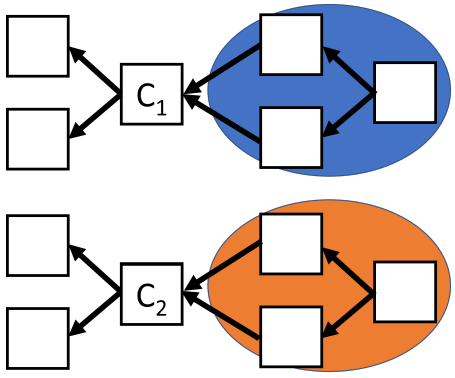

Figure 11: Conflict spamming attack

2.4.14. Issues in Milestone selection

The Coordinator uses the same TSA as any other IOTA node to determine the milestones' attachment points. One milestone is issued every minute, and if the TSA does not return tips within a specified time limit of 60 seconds, the new milestone will directly approve the last milestone [74]. This setting for the Coordinator effectively renders the milestone useless as it does not contribute to the security of the IOTA network. It is crucial to ensure that the number of transactions confirmed by milestones is high to improve the confirmation rate of the network. Also, spam transactions may flood the network during an attack, and the number of confirmed honest transactions or pending honest transactions may become a fraction of the overall transactions. In this scenario, the TSA for milestones should select tips that will confirm maximum transactions. Hence, using the regular TSA for milestones may not be optimal.

Since the Original IOTA protocol relies on the Coordinator module for security and finality, suitable Coordinator protocol improvements are needed. A suitable milestone strategy is needed to counter the spam transaction that could reduce the confirmed honest transaction. A potential solution could be that the milestone selection strategy may be run at shorter intervals of 10 seconds to increase the number of confirmed transactions. Publishing milestones at shorter intervals also has the advantage that the set of transactions considered by the milestone selection strategy will be less. Such an approach can increase the confirmed transactions in the IOTA network and improve the confirmation rate of the network, effectively countering the spam transaction attack.

Table 3 summarizes the security issues and solutions for Original IOTA given in Section 2.4.

IOTA 1.5 or Chrysalis was introduced to upgrade selected features of the Original IOTA and reduce the prevalence of attacks. The critical drawbacks of the Original IOTA were the data structure of transactions. DLTs like Bitcoin preferred a UTXO model that ensured security against double-spending and required less data for transferring BTCs. In contrast, IOTA used a heavier data structure called "bundle" to transfer tokens. Such a complicated feature created a bottleneck for broader adoption, and hence, it was necessary to have IOTA 1.5.

Table 3
Summary of security vulnerabilities and counter-measures for Original IOTA

Name of the attack	Counter-measures
Eclipsing	Yet to be resolved
Byzantine node creation	Yet to be resolved
Sybil identities	Yet to be resolved
Double spending attack	Coordinator
Parasite chain attack	TSA
Failure or Voluntary halting of the Coordinator	Yet to be resolved
Large weight attack	Coordinator
Splitting attack	Coordinator
Lazy tips or greedy attachment attack	TSA
Conflicting transactions	Coordinator
Feather attack for parasite chain	Coordinator
Privacy issues in transactions	Coin mixing, Use of Tor network
Blowball attack	TSA
Attacks on nodes	Firewall, Port scanning
API attacks	Whitelist IP, Basic authentication, Limit requests per user
MQTT broker vulnerability	Use secure TLS
Bootstrapping attack	Comparing snapshots
Reusing address attack	Modification to wallet software
Social engineering	User awareness
Lazy tips	TSA modification
Conflict spamming	Hetfield solution
Milestone issues	Milestone selection at shorter intervals

3. IOTA 1.5 or Chrysalis

The purpose of the current section is to elaborate on the features of the IOTA 1.5: Chrysalis (Section 3.1) and its strong criticism with security risks and solutions (Section 3.2). After the Original IOTA, an intermediate update was needed to optimize the mainnet before launching Coordicide. This update was named "Chrysalis" and was released to improve the usability of IOTA for all stakeholders.

3.1. Features of Chrysalis

Chrysalis consisted of algorithms for tip selection, milestone selection, white flag conflict resolution, reusable addresses, UTXO model, atomic transactions, a binary representation of transactions. The features of Chrysalis are elaborated below:

3.1.1. White flag conflict resolution approach

White flag conflict resolution approach was proposed for calculating balances. It was argued to be a more straightforward, conflict-ignoring approach that could improve the speed and efficiency of tip selection, eliminate specific network attacks, and reduce the need for reattachments. In the white flag approach, whenever there is a milestone, nodes take all the transactions since the last milestone, sort them using hash, and walk down the list of transactions [89]. Transactions that conflict with a previous transaction are dropped. The key motivation for the white flag approach was to eliminate the conflict spamming attack. With the use of the white flag conflict resolution, conflict spamming is eliminated as conflicts will be ignored. Conflicting transactions can be ignored during tip selection, making it faster to select appropriate tips.

An implication of the white flag approach is for IOTA's primary rule: "A transaction contributes to the network's security by approving two others." However, after white flag transactions can now approve conflicting or invalid transactions, additional computations will be needed to prove validity. Any transactions can approve invalid transactions without any repercussions. This could turn "one transaction approves two others" into "one transaction references two others."

Coordicide, a decentralized system, will have additional modules that affect the speed of confirming transactions. The white flag has improved the speed of the pre-coordicide implementation. However, the white flag is not a part of the coordicide and cannot speed up the network post-coordicide. The alternative would be for the developers to invest additional time and effort in making White Flag work on the post-coordicide network, thus making Coordicide even more complex. Therefore, the white flag cost months of engineering time to build, and there would be no long-term gains for post-coordicide.

3.1.2. Weighted Uniform Random Tip Selection (W-URTS)

W-URTS algorithm was proposed to enable a node to select tips that are non-lazy with the least search time to maximize the network's confirmation rate. The algorithm defines a new term *Latest Milestone Index (LSMI)* which is the index of the latest milestone. If the difference between LSMI and YTRSI is over 8, then the tip is lazy. Similarly, if the difference between OTRSI and LSMI is over 15, the tip is lazy. Finally, if the difference between OTRSI and LSMI is over 13 and below 15, the tip is semi-lazy. Lazy and semi-tips are not eligible for tip selection, and the remaining tips are non-lazy.

A node should keep a set of non-lazy tips. Every time a node is asked to select tips to be approved, it will pick randomly from the set. A tip should not be removed from the tips set immediately after it was selected to be approved to make it possible for it to be re-selected. Such a step, in turn, makes the Tangle wider and improves synchronization speed. A tip is removed from the tips set if a certain number of transactions approve it or if a certain amount of time is passed [95].

3.1.3. Edwards-curve (Ed25519) signature scheme

The Ed25519 signature scheme has been added to the network, replacing the current W-OTS signature scheme. Table 4 gives certain features of the Ed25519 scheme [11]:

The W-OTS scheme had certain disadvantages compared to Ed25519, such as the size of the signatures, the time needed to validate one signature, and reusability. The W-OTS scheme should generate only one digital signature as it reveals an unknown amount of the private key and can lead to a brute attack on the private key. Secondly, the digital signature created is immense, requiring 1300-3900 bytes storage. Finally, the hashing function needs to be executed 702 times to validate one signature in the default setting,

Table 4
Ed25519 and W-OTS computation cost

	Ed25519 [11]	W-OTS [93]
Signature verification time (in cycles)	273364	70 * 273364
Batch signature verification (in cycles per sign)	134000	70 * 134000
Key generation (in cycles)	87548	702 * 87548
Signing speed (in cycles)	6000	702 * 6000
Signature size (in bytes)	64	1300-3900
Key size (in bytes)	32	256

leading to significant system overhead even on powerful hardware.

An Ed25519 signature scheme allows the protocol and clients to run more efficiently on established hardware. Unlike W-OTS, the Ed25519 signature scheme also allows for the re-use of private keys and introduces reusable addresses to the protocol. This change also dramatically reduces the transaction size, saving network bandwidth and processing time [92]. In summary, the signature scheme switch meant a reduction in communication overhead as the signature size reduced [93].

3.1.4. Atomic Transactions

Atomic transactions were proposed to update the current, complicated bundle construct instead of more straightforward atomic transactions. The aim was to provide results in much simpler development and more adaptable and maintainable code of the core software. In addition, atomic transactions were argued to reduce network overhead, transaction validation, and signature verification load and improve spam protection and congestion control.

Currently, IOTA organizes transactions into a bundle that consists of multiple transactions that rely on each other. For example, one bundle can have a transaction that withdraws tokens from a sender's address and another transaction that deposits tokens into the receiver's address. Additional transactions carrying the W-OTS signatures could be part of the same bundle. The bundle concept has proven to be time-consuming, with several issues as well: Since the data making up the bundle is split across multiple transactions, it complicates the validation of the entire transfer. Instead of immediately telling whether a bundle is valid or not, a node must first collect all parts of the bundle before any actual validation can happen. This increases the complexity of the node implementation. Furthermore, reattaching the tail transaction of a bundle causes the entire transfer to be reapplied. Finally, PoW has to be completed for each bundle transaction, making transferring tokens a slow and computation exhaustive process.

To overcome the drawbacks mentioned above, IOTA Foundation proposed an atomic structure with support for Ed25519 (and thus reusable addresses). Additionally, it would support adding new types of signature schemes, addresses, inputs, and outputs as part of protocol upgrades, self-contained, as in being able to validate the transaction immediately after receiving it, enable UTXO as inputs instead of an account-based model [50].

3.1.5. Switch to UTXO Model

Currently, IOTA follows the account balance model, which introduces complexity in identifying double-spends, and as IOTA moves towards decentralization, it would need a fast and efficient alternative to identify double spends. Therefore, the UTXO model was proposed by IOTA Foundation as a part of Chrysalis. The unspent transaction output (UTXO) model defines a ledger state where balances are not directly associated with addresses but with the outputs of transactions. In this model, transactions specify the outputs of previous transactions as inputs, which are consumed to create new outputs. A transaction must consume the entirety of the specified inputs. Using a UTXO based model provides several benefits over an account balance-based model, such as parallel validation of transactions and easier double-spend detection since conflicting transactions would reference the same UTXO. Figure 12 shows the UTXO model in IOTA where 100i and 50i are transferred from the genesis block to two addresses. These two addresses are used in transaction 1 as inputs to transfer 150i to address "output 1" which is unspent. The unlock block in transaction 1 holds the signature(s) unlocking the input(s) [50].

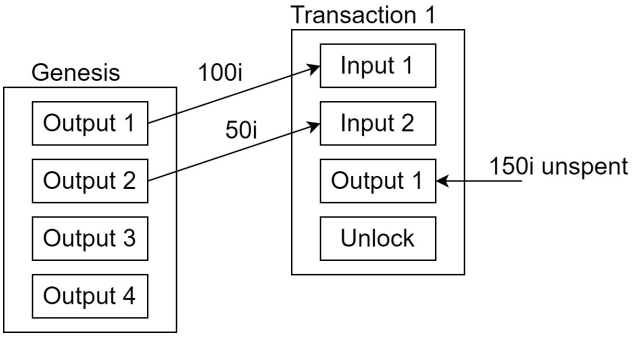

Figure 12: UTXO Model of IOTA

The new UTXO transaction format shall become the core data type within the IOTA ecosystem, replacing the previous transaction format. Thus, all client libraries, blueprints, PoC, and IOTA applications must be updated to use the new format. Additionally, these changes are breaking, meaning that all nodes must upgrade to further participate in the network.

3.2. Attack vectors and counter-measures for Chrysalis version

Further, we discuss the issues and criticisms of the Chrysalis update that could lead to insecurity in the network and preventive remedies for the same.

3.2.1. Spam attack - White flag issues

White flag resolution is valid only for the pre-coordicide version of IOTA and is thus a temporary solution. As the white flag removes the problem of conflicting transactions, the importance of the milestones is reduced. In other words: The Coordinator's milestones can now approve anything, and it is up to the nodes to figure out which transactions are valid and which ones are not. This would create a 100% confirmation rate because every transaction (valid or not) can now be confirmed. This also removes the need for reattaching and promoting transactions, as there are no invalid branches anymore. It also simplifies the tip selection process because every tip can be attached to, valid or not [19]. The white flag approach has several implications. Primarily it would reduce dependence on the Coordinator as the Coordinator can approve any transaction, including invalid ones. Hence, to prove that a specific (non-milestone) transaction is valid, it is no longer sufficient to provide the "path" to its confirming milestone but instead all transactions in its past cone. The milestone is no longer key for security.

3.2.2. Denial of service attack - Low-powered devices issues

The IOTA Foundation launched their full node software named Hornet written in Go and capable of installing and executing on Raspberry Pis. However, low-powered devices running full node software such as Raspberry Pis could not handle the high transactions per second [18]. IOTA Foundation observed that Raspberry Pi's were dropping out of the network (losing sync) at 150-400 messages per second which are about 35-100 bundles per second.

Hence, it is recommended to prefer a machine with an excellent configuration to handle a high rate of transactions per second. Nodes need enough computational power to run reliably, including the following minimum requirements: A dual-core CPU, 2 GB RAM, and SSD storage [67].

3.2.3. Data availability - Production systems

Data or zero-value transactions do not affect the ledger balance. However, these are important as they could contain critical data shared by servers. Tangle being a DLT does not guarantee confirmation to transactions, and they could be dropped in the event of node crashing [20]. Even protocol issues such as if the transaction was broadcast with UDP might not reach the nodes due to heavy traffic congestion in the network. Secondly, as nodes must prune data periodically (default is seven days for mainnet), the data might not be available at the node after seven days.

Legal challenges related to General Data Protection Regulation (GDPR) compliance also may exist for production systems that use Tangle. Articles 16 and 17 of GDPR state that data can be modified or erased where necessary to comply with legal requirements. The Tangle, however, does not comply with GPDR articles as it provides data integrity and increases trust in the network [78, 12].

3.2.4. Quantum computation attack

Quantum computation attack refers to the ability of a device to execute the Shor's quantum algorithm proposed in 1994. The Shor's algorithm is designed for finding the prime factors of an integer in polynomial time $log\,N$ (N is the size of the input integer) demonstrated that public-key cryptography schemes such as Finite Field Diffie-Hellman key exchange and Elliptic Curve Diffie-Hellman key exchange could be broken if a quantum computer with sufficient specification existed [17]. Public key cryptography is based on the assumption that factoring large integers is computationally intractable. As far as is known, this assumption is valid for classical (non-quantum) computers; no classical algorithm is known that can factor integers in polynomial time. Hence, if and when quantum computers are available, many classical public-key cryptography schemes broadly used would become compromised. Existing cryptocurrencies have not so far employed quantum-resistant signature schemes relying instead on more common non-quantum-resistant solutions [93]. As previously discussed in Section 2.4.10, W-OTS has major disadvantages, which have impeded its large scale institutional adoption. As the quantum threat is not within sight yet, IOTA adopted the Ed25519 signature scheme, a popular yet, in theory, with the drawback of being breakable by a quantum computer running Shor's algorithm.

Another issue highlighted by S. Popov in [61] is about a sufficiently large quantum computer that could be very efficient for handling problems that rely on trial and error to find a solution. In IOTA a good example of such a problem is the number of nonces that one needs to check in order to find a suitable hash for issuing a transaction. In IOTA, the number of nonces that must be checked to issue a transaction is 6561 [61]. Theoretically, using quantum computers, the nonces needed to be checked become $\mathcal{O}(\sqrt{N}) = \sqrt{6561} = 81$. Thus, quantum computers can issue transactions 81 times faster than a regular computer. Hence, the attacks such as parasite chain attacks and double-spending attacks become easier if a quantum computer is used.

3.2.5. Privacy vulnerabilities in transactions

An attacker may use UTXO transactions of IOTA to disclose the identity of users. The Tangle is public, distributed, and transparent, and anyone that a user transacts with can see the user's total balance and parts of their transaction history [82]. Taint analysis is also possible on the IOTA transactions [82]. Taint analysis aims to quantify associations between addresses. Figure 13 shows the taint of address two will be on address four and any address to which transacts with address 4. If any of the four addresses can be discovered, the other three identities could be compromised.

When a transaction is to be made, a single address owned by the sender may not have the entire amount. In such a case, the sender uses multiple addresses controlled by

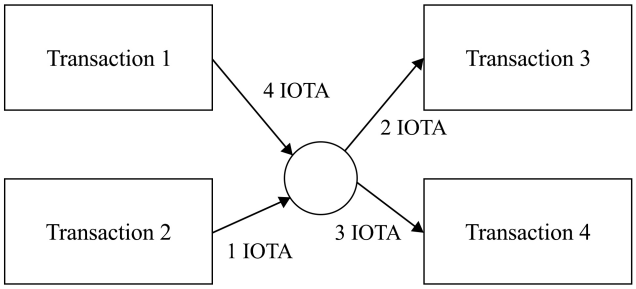

Figure 13: Taint analysis

him/her. Inevitably a change address also shall be created [82]. It is possible to link the multiple change addresses to a single account, compromising privacy. The MAM is another feature on IOTA that can also compromise privacy. In MAM, the message occupies the sender's signature field. The default setting allows messages to be unencrypted. This makes messages visible to users. However, to counter the integrity and confidentiality attacks on the MAM, it is possible to encrypt messages, ensuring confidentiality, integrity, and authentication.

As IOTA gains acceptance in the crypto-currency market, the trading shall increase. The trading of IOTA with other fiat currencies requires correspondence with exchange houses. Such exchanging houses store identifiable information of users due to regulatory requirements. A privacy flaw in the IOTA protocol is that connecting to full nodes to make wallets transactions is required. Specific full nodes do not allow connections from Tor IP, making the anonymous publication of transactions difficult. Additionally, IOTA uses manual peer discovery, which requires a full node to maintain a static IP. So it is more difficult to route transactions anonymously.

Coin mixing to improve anonymity is a tried and accepted approach to obtain confidentiality in Bitcoin and could be used even for IOTA. L. Tennant [82] suggested CoinJoin for improving anonymity of IOTA token transfer. However, CoinJoin is observed even in other DLTs. The following analogy explains CoinJoin: Alice wants to transfer 10 IOTA to Lewis, and Bob wants to transfer 10 IOTA to Carol. Any DLT is transparent, and details of the transaction shall be visible to everyone. To avoid deanonymization, Alice and Bob transfer to CoinJoin, which is a coin mixing service. Then CoinJoin would transfer to Carol and Lewis. The coin mixing method leads to Alice and Bob's anonymity as IOTA tokens transferred to Lewis and Carol will be traced to CoinJoin in an audit trail. Figure 14 illustrates the use of CoinJoin for improving the anonymity of transactions in IOTA. However, coin mixing is most efficacious when the amount of IOTA transferred by users, who wish to participate in coin mixing, is the same. Otherwise, it is trivial to determine the senders. The analysis of Coin mixers in Bitcoin networks concludes that an incentive coin mixer will not emerge unless there is an incentive coin mixer. Additionally, the Coin mixing service knows all participants' identities, so the compromise to privacy still exists. Coin join transactions are distinguishable compared to everyday transactions due to the high inputs and outputs. So, coin joins can be easily identified on the ledger. Coin mixing is preferred for removing the taint from coins. As multiple users will depend on a single mixer, the mixer absconding with the IOTA tokens will inconvenience many IOTA users.

Table 5
Summary of security vulnerabilities and counter-measures for IOTA 1.5

Name of the attack	Counter measures
Spam attack	Mana reputation system
Denial of service	High configuration machines
Data availability	Using permanodes
Quantum computations	Existing resources cannot launch such attacks.
Privacy	User awareness

Table 5 summarizes the security issues and solutions for IOTA 1.5 given in Section 3.2.

Chrysalis improved the TPS of the network, yet for creating a scalable ledger for IoE with limitless transaction throughput, the Coordinator module would have to be removed. Hence, the need for the third update of IOTA, known as "Coordicide".

4. Security issues and solutions for IOTA 2.0 or Coordicide version

This section describes all the attacks and solutions available in the literature to safeguard the beta version of the Coordicide version. The solutions are organized in the same order as the vulnerabilities in the previous section. Due to the nascent state of IOTA research, in addition to scientific publications, we have included security solutions that are from sources such as blogs, discussion forums, the official IOTA discord server and the research repository of the IOTA Foundation [64].

4.1. Analysis of the beta version of the Coordicide protocol

Coordicide protocol was necessary as the Coordinator became a bottleneck in the Original IOTA. Coordicide aimed to provide decentralization, to create permissionless DLT, to avoid single point of failure, and to remove performance bottleneck. Although, the concept of "Stars" could mitigate the centralized point of failure of the Coordinator. When writing this paper (September 2021), "Stars" was yet to be deployed on IOTA's mainnet. However, in the long run, the Coordicide protocol is the most suitable option as it will remove the need for Coordinators as defense mechanisms. IOTA Foundation had always advocated the Coordinator as a bootstrapping mechanism rather than a long-term solution. Hence, the Foundation moved to a reputation-based approach called the Coordicide version of the IOTA protocol. Since the

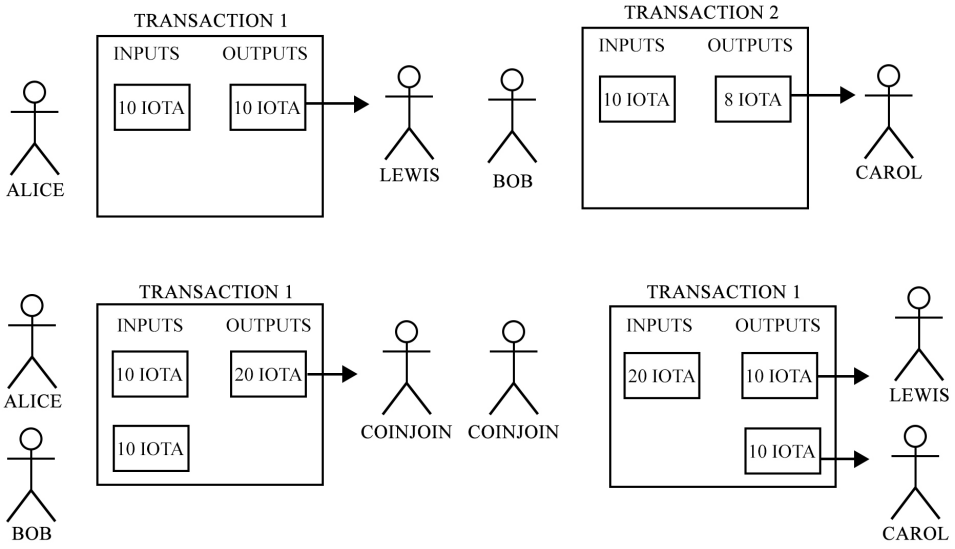

Figure 14: Coinjoin schemes.

Coordinator and milestones remain critical aspects of the IRI software, removing them posed significant challenges. Currently, Coordicide is under testing phase, and its impact needs to be studied in detail. So this section analyses the additional critical modules of the Coordicide version. These additional modules are mana mechanism, peering system, rate and congestion control mechanism, and decentralized consensus protocols.

4.1.1. Peering

Figure 15 illustrates the peering process a new node performs whenever it joins the IOTA network. For example, Node *A* has newly entered the IOTA mainnet, and the first thing *A* needs is to connect to neighbors. There are special entry nodes in IOTA networks that support newly joined nodes with IP addresses of other nodes. Node *A* can then choose its neighbors with the help of entry nodes. The node software manages neighbors of a node and may remove lazy neighbors. When a node receives a transaction, it communicates this transaction by gossip protocol to its neighbors. This "gossiping" continues till that transaction reaches all nodes in IOTA.

In IOTA, the entry of a byzantine node may affect the peering process. We envisage the following attack scenarios:

Scenario #1 - Byzantine entry node: In IOTA, the only entry barrier for public entry nodes is a domain name with both an A (IPv4) and AAAA (IPv6) record. The heartbeat protocol monitors public entry nodes, which are also mentioned on official documentation of IOTA. However, as IOTA expands, the possibility of a byzantine entry node will grow. Moreover, IOTA does not regulate private Tangles, and node software currently offers less protection against byzantine entry nodes in private Tangles;

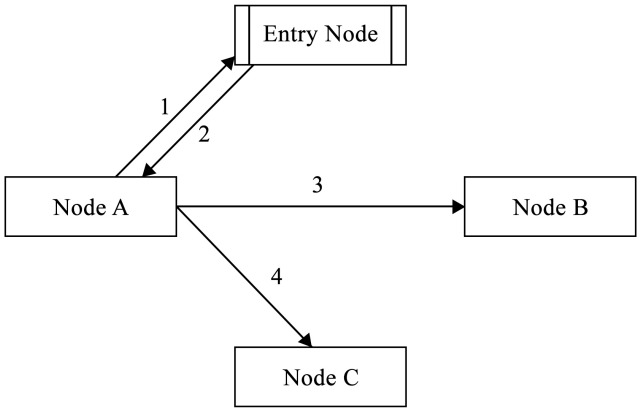

Figure 15: Peering process

Scenario #2 - Eclipse attack: A byzantine node may join IOTA mainnet and obtain the IP address of other IOTA nodes by requesting an entry node. Then a DoS may be launched against another IOTA node or an entry node;

Scenario #3 - Sybil attack: As IOTA is free, even DDoS attacks using multiple Sybil identities are convenient. A Sybil attack can be made against a particular node by refusing to forward transactions to/from it.

4.1.2. Mana system

Implementation of the Mana system has three aspects:

1. Pending mana: An address consisting of tokens generates pending mana;
2. Mana: When we spend tokens from an address, pending mana is converted to mana and pledged to a node which writes the transaction to Tangle;
3. Decay: Both mana and pending mana decay proportionally to their value.

The amount of mana in IOTA remains constant, and so in effect, mana is neither created nor destroyed and only can be transferred from one node to another. There will be a race between nodes to keep their mana high. Higher mana means the higher reputation of the node. As a reward for higher reputation, nodes send more transactions to IOTA and get higher voting power as oracles in assemblies. If a node processes more transactions or transactions of higher value, its mana increases. Following security scenarios may occur:

- A byzantine user may create transactions transferring tokens to addresses controlled by self. A node created by the user routes these transactions. Such a scheme may artificially inflate mana of a node;
- An attacker with multiple byzantine nodes may block transactions to/from honest nodes affecting the mana honest should have obtained.

The pending mana an address x holding S tokens has at time t is given by $m_x(t)$ [69]. For generation rate for pending mana α and the decay rate coefficient for mana and pending mana γ, $m_x(t)$ is calculated as follows:

$$m_x(t) = m_x(0)e^{-\gamma t} + \frac{\alpha S}{\gamma}(1 - e^{-\gamma t}). \quad (1)$$

When a node processes such an address, the pending mana is converted to mana and pledged. Assuming that $m_x(0)e^{-\gamma t}$ is zero then $m_x(t)$ will be simplified. So for an attacker to change a node's reputation, it has to block transactions to it. The reputation of the node will decay. IOTA proposes the Mana system with the philosophy of "difficult to gain and easy to lose". In IOTA, the protocol will implement it as a feature that allows reassigning a granted mana to another node. However, it may be challenging to decide who shall decide on reallocation and mana reassignment in a decentralized scheme.

4.1.3. Private Tangles

Figure 16 illustrates the architecture of a multi-node private Tangle and a public Devnet Tangle. Both architectures are the same except that public Devnet has a load balancer that allocates transactions to nodes to avoid excessive pressure on a node. Nodes gossip using TCP 15600, whereas external communications with clients, load balancers, or COO happen using TCP 14265. IOTA Foundation gives limited attention to providing security solutions to private Tangles as they are meant for experimentation or internal use for organizations.

Following security issues and solutions are envisaged:

- Use of light wallet to send transactions is possible in private Tangles. Light wallets send data over HTTP and so snooping over transactions is possible. Trinity wallet could be used to overcome this issue;
- The "one-command Tangle" uses a pre-computed Merkle tree with a public seed so anyone who has the URL of your node can use the compass seed to take over the private Tangle. Alternatively, building a private Tangle and setting a random seed should be preferred;
- Limited nodes in private Tangle may lead to a node going offline in the event of high incoming transactions. (i) Whitelist IP address, (ii) Implement basic authentication, (iii) Limit API requests from each user could be solutions. Alternatively, we may deploy a load balancer.

4.1.4. Congestion and rate control

Network congestion is an undesirable effect in DLTs. Congestion arises due to frequent message passing between nodes. This message passing is inevitable in a DLT. Messages passed in DLT are: (i) "gossips" to transfer transactions to nodes, (ii) transactions request to node, (iii) API calls by clients, (iv) peer discovery messages, and (v) heartbeats for verifying the status of nodes. Attackers can exploit the current protocol stack of IOTA to create nuisance in the network. The issues are as follows:

- Attacker can create fake transactions which may be "gossiped" by nodes throughout the network to create congestion;
- Attackers may use high-speed devices such as FPGAs to send continuous transactions in the IOTA, and low-end IoT devices may face hurdles in sending transactions.

In the IOTA congestion control algorithm, each node has a scheduler that iterates queues corresponding to different node IDs. Each queue stores messages belonging to a single node ID. If the number of messages in a specific queue exceeds a certain threshold (proportional to q/n), where q is the queue length, and n is the node's reputation. Then this queue, which may correspond to the malicious node, will be blocked. This action is performed locally in a node. When a node is blocked, all messages received by this specific node ID are dropped for a particular time. Eventually, all neighbors will block the malicious node, effectively eclipsing it. Honest nodes are at less risk of being blocked because they generate messages according to the rate-setting algorithm, which considers the current congestion status of the network. In a network with high latency, honest nodes may inadvertently get blocked if they send more messages than is allowed. An attacker may also block an honest node to isolate it from the network.

Currently, IOTA follows PoW for spam protection. However, certain devices send PoW requests to nodes that perform PoW on their behalf. For such devices, setting a high difficulty will not mitigate spam [21]. Moreover, the high difficulty might make it difficult for ordinary IoT devices to perform PoW and make transactions. Coordicide protocol suggests an adaptive PoW computation to resolve spam. The adaptive PoW states that, to send a transaction at time t, a

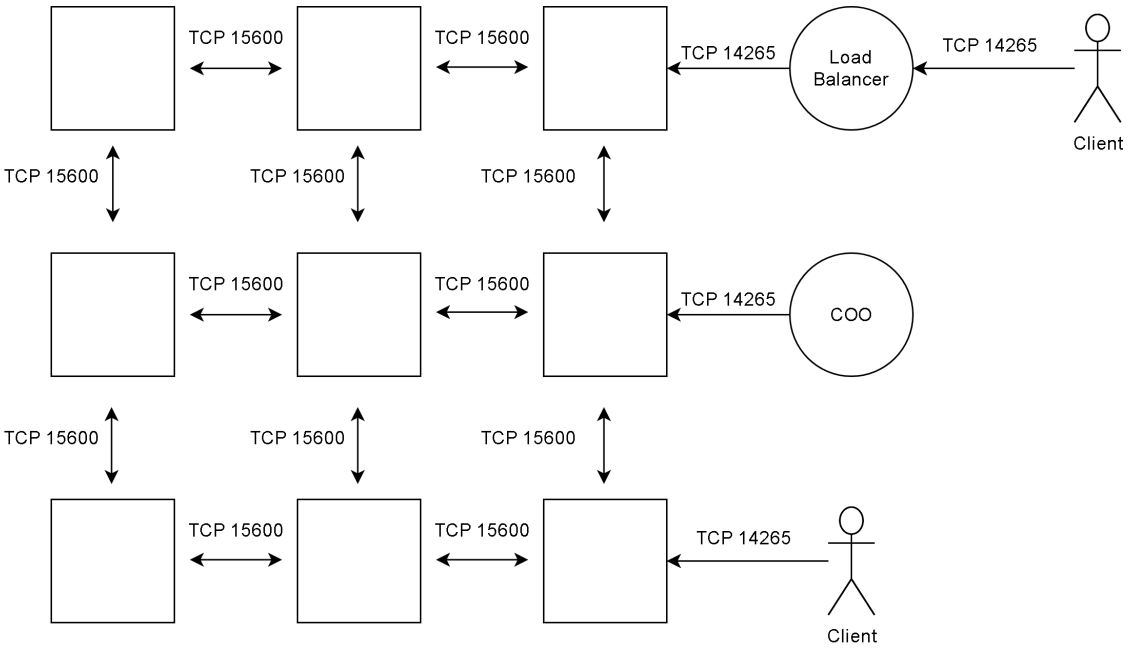

Figure 16: Common architecture of Private Tangles and Public Devnet of IOTA.

node i must perform PoW with difficulty $d_i(t)$ given by the following equation [69].

$$d_i(t) = d_0 + \lfloor \gamma \cdot a_i(t) \rfloor \,, \tag{2}$$

where,

- d_0: the minimum difficulty of PoW;
- $\gamma_i \in [0, 1]$: the rate to adjust difficulty; This rate depends on mana owned by node i;
- $W > 0$: time window;
- $a_i(t)$: number of transactions issued by node i in time interval $[t, W.t]$.

When a node receives a transaction from i with difficulty d_i it checks $d_i \geq d_0 + \gamma_i * r_i(t)$. Here, $r_i(t)$ are the number of transactions received from i in time W. If the condition is satisfied, then the transaction is forwarded. In such cases, each node will have to maintain additional meta-data for verifying transactions introducing storage and computation cost on a node. Therefore, adaptive PoW may introduce an additional resource burden on nodes.

4.1.5. Consensus Mechanisms

Nodes in the IOTA network need to have a mechanism to come to a *consensus* on conflicting transactions. In the original IOTA, the consensus is achieved through a biased random-walk TSA. In case of a conflict, the TSA would leave all but one of the conflicting transactions. In such a mechanism, conflict resolution shall be slow and may lead to a "wrong" branch getting orphaned, requiring many reattachments. For the Coordicide protocol, a novel consensus mechanism named "Shimmer" was proposed. In Shimmer, a node would query a subset of other nodes about their opinion on a particular transaction. The opinion of nodes can be either 0 (reject) or 1 (accept). Then it would form its own opinion about the transaction based on *voting* the opinions of the queried nodes. After the vote, the node either would "like" or "dislike" that transaction. Two voting mechanisms described in the Coordicide white paper are: Fast Probabilistic Consensus (FPC) and Cellular Automaton (CA) [69]. An ideal consensus mechanism should be leaderless, have low communication complexity, have low false positives and false negatives, and be robust to adversarial nodes working to delay consensus.

- FPC: Loosely speaking, this mechanism states, "If the past l rounds result in at least q proportion of my queries returning 1-opinions then my opinion is 1". The mechanism has the following steps [68]:

  Step 1: Each node can decide its initial opinion about a transaction by following any reasonable rule. For example, if a node sees a transaction at time t, it checks that it does not conflict with any prior transaction. Also, any subsequent transaction received during the time interval $[t, t + \Delta]$ does not conflict. If both these conditions are satisfied, the initial opinion is set to 1.

  Step 2: Then in the first round, a node would query k nodes and record the number of 1-opinions ($n(j)$) it receives. It decides its opinion as 1 if $k^{-1}n(j) \geq X$, where X is a random number and $\sim U[0.5, 1]$. Otherwise it decides 0.

  Step 3: For the subsequent rounds, a node would query k nodes and record the number of 1-opinions ($n(j)$)

it receives. It decides its opinion as 1 if $k^{-1}n(j) \geq X$, where X is a random number and $\sim U[0, 0.5]$. Otherwise it decides 0.

If a node maintains its opinion during l rounds, it becomes final. Due to propagation delay, not all node responses may be received. Hence, a node may query more nodes and only accept the first k responses within a time limit. For low communication complexity, $k \geq 50$ and $k << n$, where n is total nodes in the IOTA network. Nodes follow a local stopping strategy, and when all nodes stop querying, the mechanism terminates. The FPC mechanism relies on random numbers, and Coordicide may decide to use a trusted entity to provide random numbers or create a decentralized solution (nodes may be allowed to generate the random numbers using generator protocols) [22].

- CA: Every node decides its opinion on a conflict by querying its direct neighbors and adopts the majority opinions of its neighbors. When a node queries its direct neighbor, the response should contain a "proof" which includes the opinions of the neighbors' neighbors.

FPC and CA are not mutually exclusive and can be used together for building a robust protocol. Both FPC and CA assume that the nodes sign their responses to facilitate authentication. A termination criterion is introduced for both FPC and CA in which all nodes shall maintain a counter variable that is incremented by one whenever there is no change in the opinion. Once this variable reaches a threshold, the protocol halts. After the threshold is reached, the node only replies to queries and does not send new queries. Also, there is a cap for the number of rounds by each node, after which the mechanism stops.

4.2. Attack vectors and counter-measures for Coordicide version

Coordicide protocol has introduced several functionalities and altered the core IOTA protocol. Certain attack vectors that could be possible in the current protocol version are mentioned.

4.2.1. Qubic protocol attacks

Sybil attack on Qubic protocol involves a single oracle that can impersonate multiple oracles to get a larger reward. A second attack is the classroom attack, where oracles can copy results from other oracles without verifying.

Qubic protocol faces Sybil and classroom attacks. A practical solution to the Sybil attack is using a weighted voting scheme. Each oracle will be assigned votes based on the computational resources it contributes. A node must conduct a resource test every time a new oracle joins and assign weights to each oracle. Classroom attacks can be mitigated by keeping the oracles' responses until all oracles send the data [5].

An additional counter-measure for protection against Sybil attack is the Mana system. Mana is a reputation value given by a user to a node. Coordicide implementation proposes mana to be equal to IOTA tokens transferred. The more mana the node has, the higher its reputation [69]. By calculating the mana of a node, other nodes can decide its reputation. A node can calculate mana for other nodes using node software. This reputation mechanism can be used to decide on whether to keep a node as a neighbor, send/receive transactions to/from a node, and prevent a node from overwhelming the network.

4.2.2. Spam attack and dust attack

PoW is a mechanism proposed by IOTA to prevent a single node from spamming the Tangle with transactions [21]. However, it is allowed for an issuer to perform PoW on full nodes remotely. A spammer can use multiple full nodes to issue spam transactions on Tangle without any costs or hardware. Spammers may also use specialized hardware (e.g., FPGAs) to produce many transactions and congest the network. A variation of the spam attack is the "dust" attack. It is a memory exhaustion attack targeting nodes in which an attacker creates transactions transferring minuscule amounts of tokens to different addresses. A full node requires 49 bytes to store an address in its database and 8 bytes to store the amount. If 10^9 IOTA is distributed to 10^9 addresses, 57 GB ($57 * 10^9 bytes$) storage would be exhausted in the full nodes.

For protection against Spam and dust attacks, IOTA advocates PoW so that any node which wants to send a transaction or a message must use computational resources. PoW ensures that spamming will be costly for the attacker [69]. Adaptive PoW [86], Verifiable Delay Function (VDF) [3], or reputation-based congestion control mechanism [26] are proposed to mitigate spam attacks in the beta version of the Coordicide protocol. A minimum amount that can be transferred will be decided to mitigate dust attacks, and nodes will ignore all transactions transferring tokens below this threshold. Also, an address must hold a minimum balance of tokens to make a transfer. If a token transfer reduced the tokens in an address below the threshold, the nodes would ignore such transactions. Such a mechanism would make dust attacks expensive and ensure that IOTA could still handle microtransactions.

4.2.3. Attacks on private Tangles

Operating a private Tangle is the entity's responsibility. IOTA Foundation provides limited support on the node software. The node software recommended in private Tangle version 1.0 is IRI which was deprecated for use in the mainnet. The IRI configuration for private Tangles is the same as that on Devnet, i.e., MWM = 9, milestone ticks = 60 seconds. In private Tangles, the attacks shall be from internal members than external ones. Following attack scenarios are envisaged:

- IRI uses a seed phrase to create public/private keys for signing bundles. If the seed is compromised, an

attacker can send fraudulent milestones to disrupt the private Tangle.

- If the node to which IRI is connected becomes compromised, it could lead to situations such as:
  - Attackers transactions will be preferred over the regular tips.
  - Double spends will be created, causing inconsistent milestones. IRI nodes will not accept this milestone, and transactions will not be confirmed.
  - Gossiping milestone transactions will be stopped by the rest of the network, causing Tangle to freeze.
- IOTA Foundation also provides "one-command-Tangle", a private Tangle in a docker container that starts with a single command. IOTA Foundation recommends "one-command-Tangle" only for experimentation. This private Tangle uses deprecated IRI node software and a fixed seed. Hence, exposing public applications will give attackers easy access to Tangle.

4.2.4. Orphanage attack

Coordicide IOTA would be utilizing a TSA named Restricted Uniform Random TS (RURTS). In RURTS, a transaction will be allowed to attach to tips not older than itself by Δ. Δ refers to the timestamp difference between the transaction and a tip. This introduces the possibility that some transactions may remain as tips forever, i.e., become orphaned if they do not obtain an approving transaction within a particular time interval [57]. To improve the confirmation rate of the network, it must be ensured that the likelihood of this event happening remains negligible. A malicious node could spam the network with transactions that do not attach to legitimate transactions in an adversarial environment. In times of under-utilization in the network, the adversary may obtain a transaction issuance bandwidth close to 1 allowing such attack. This attack may be detected by monitoring the number of tips in the network for sudden increases.

The adversary has to obtain a sizeable proportion of the transaction issuance bandwidth for this attack to succeed. In the Original IOTA, the adversary can corner the bandwidth by investing PoW. However, in the Coordicide version, the mana system may increase the difficulty of the attack. If it is assumed that the attacker has sufficient mana, then the node software could include a setting that could dynamically adjust the transaction issuance rate [57]. If the nodes detect an odd number of tips in the network, they may increase the transaction issuance rate to a certain proportion of their allowance. With this approach, the adversary may not obtain the required bandwidth for the attack.

4.2.5. Adversarial attacks on the consensus mechanisms

The FPC and CA mechanisms are observed to suffer from three major failures [22]:

1. Termination failure: if the nodes reach the cap for the number of rounds without correct opinion on the conflicting transaction
2. Agreement failure: if not all the honest nodes reach the same opinion on the conflicting transaction
3. Integrity failure: if the final opinion of the honest nodes is incorrect on the conflicting transaction

These failures result from adversarial attacks from three types of adversaries present in the IOTA network:

1. Cautious or covert: maintains the same opinion in the same round.
2. Berserk: maintains different opinions in the same round.
3. Semi-cautious: few nodes controlled by the adversary are berserk, and few are cautious or may even remain silent.

The FPC mechanism is vulnerable to attacks resulting in delayed consensus or failure if $\geq$ 50% cautious adversaries or $\geq$ 33% berserk adversaries, or $\geq$ 38% semi-cautious adversaries are present during any round [68]. Additionally, in the FPC mechanism, there is a possibility that the adversary can hijack the random number and be able to delay consensus. It is argued that with randomization, the adversary might not be able to control the honest nodes, yet it can delay the termination of the protocol or reduce its integrity [22]. The CA mechanism is vulnerable to eclipse attack as the adversary may create multiple Sybil nodes due to no entry barriers in the Coordicide IOTA. Both types of adversaries (covert or berserk) can launch eclipse attacks. In IOTA, the adversary may create many Sybil identities and continuously query honest nodes for opinions. As no entry barrier is available in the Coordicide IOTA, Sybil identities can be used to overwhelm an honest node with queries resulting in a DoS or DDoS attack or "Delay" in-service attack where the turnaround time for queries of honest nodes may increase. Calculating the communication cost of FPC and CA, if each round requires $(k - 2k)$ messages to be sent per node and if l rounds are assumed, then total messages to be sent are $(l * k - 2 * l * k)$ for a single conflict. Hence, for all nodes in the network, reaching consensus on a single conflict will require time and may load the network creating latency.

Additional modules can be introduced in FPC and CA mechanisms to counter the attacks. If any honest nodes exchange information of their query history, then they can isolate and penalize any adversary present in the network. However, the *vanilla* versions of FPC and CA may need to adopt additional measures which permit query history exchange to enforce and detect adversaries.

4.2.6. Attacks on Cyber-Physical Systems (CPS)

It is also worth discussing cyber-attacks that affect critical infrastructure, especially CPS [27]. Research papers discussing IOTA provide little information about such attacks even though IOTA infrastructure may face such attacks too. Hence, for the convenience of the readers, we have

identified and described attacks on CPS that are relevant for the IOTA network.

Attack #1 - Replay attack on IOTA nodes: Replay attack, also known as playback attack, is a network attack in which the adversary captures a valid data transmission such as a hash of a password and uses this data to gain access to a device [83]. Figure 17 shows one scenario where such an attack can be carried out. While the administrator logs into the IOTA node, an adversary might capture the password's hash. The adversary may then use this hash to access the IOTA node.

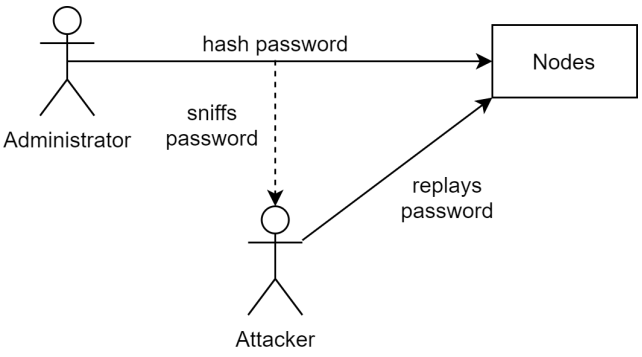

Figure 17: Replay attack

Attack #2 - Password guessing attack on IOTA nodes: An adversary attempts possible variations of passwords to figure out the correct password. The administrators of IOTA nodes would be vulnerable to this attack. The existing password may be cracked using brute force [83].

Attack #3 - IOTA impersonation attack: In the classical node impersonation attack, an adversary impersonates a server or a device in order to obtain essential and confidential data. In an IOTA network, an adversary may create byzantine nodes and steal confidential information from clients that request such nodes for remote PoW. Additionally, the adversary may also be able to decipher client identities by linking public keys with clients' information and would then be able to identify client transactions, client behavior, or even clients' net worth.

We advocate employing the counter-measures mentioned below to mitigate adversarial attacks on the CPS and protect it.

- Securing IOTA nodes against Replay attacks: One security measure against replay attack employs two-factor authentication measures. The multiple levels of authentication should employ different channels of communication. Hence, if one channel is compromised, the other authentication channel shall prevent an attacker from getting unauthorized access.
- Securing IOTA nodes against Password guessing attack: In addition to two-factor authentication, there should be a limit to the number of password attempts that an entity can make before the access is locked and the administrator notified.
- Securing IOTA nodes against impersonation attack: Clients requiring Remote PoW should not rely on a single node for broadcasting the transactions and forward the transactions to different nodes to avoid a single point of vulnerability for the client.

Table 6
Summary of security vulnerabilities and counter-measures for IOTA Coordicide version

Name of the attack	Counter-measure
Qubic protocol attacks	Weighted voting scheme and hiding oracles' responses
Spam attack and dust attack	Adaptive PoW, VDF, reputation-based congestion control mechanism
Attacks on private Tangles	Safeguards in IRI
Orphanage attack	Mana system
Adversarial attacks on the consensus mechanisms	Exchange query history to isolate and penalize adversary
Replay, Node impersonation and Password guessing attack	Two factor authentication, Protocol changes

Table 6 gives the summary of security issues and solutions for Section 4.

5. Open Research Problems

Due to the rapidly changing specifications of the IOTA protocol, there is broad scope for further research on security aspects, protocol optimizations, or functionality [63]. Open RQ is listed to guide future research in IOTA. The main idea for this section is to ensure that research remains focused on specific aspects to service the IOTA community's needs effectively.

5.1. Alternative mechanisms for spam prevention in the Tangle

Currently, the IOTA network uses PoW as a spam-prevention mechanism. However, with specialized hardware such as FPGAs, users can complete PoW within a short duration, making PoW ineffective for reducing spam. However, it is worth mentioning that the flexibility to perform `attachToTangle()` operations in a very agile way is one of the aspects that attract users, especially in the IoT domain. As a result, there is a need for alternative mechanisms to prevent spam and, at the same time, maintain the flexibility needed for IoT operations. IOTA Foundation has recognized this as one of the top priorities for further research. Methodologies for spam prevention put forth should answer the queries such as follows:

RQ #1: Is the utilization of alternatives to Adaptive PoW necessary and sustainable?

RQ #2: Would the enforcement of admission control strategies such as filtering transactions by node reputation be sufficient to control spam?

RQ #3: Can it be possible to have spam control and at the same time continue the agility of `attachToTangle()` operations?

Research is ongoing to replace Adaptive PoW with a sustainable mechanism known as VDF. VDFs are special functions that are easy to verify yet difficult to evaluate even if multiple parallel processors are used. Hence, VDFs avoid mining races, make utilization of dedicated hardware inefficient and solve the unfairness between fast and slow nodes [3, 4]. Acceptance of VDF by the IOTA community is dependent on whether it can sufficiently answer the questions mentioned above.

5.2. Avoid congestion in the network layer by optimizing networking

Nodes are responsible for implementing the IOTA core protocol and thus ensuring that all participants in the network follow standards prescribed for joining the network. Nodes use their bandwidth, memory, and computing power to perform their routine tasks, such as processing transactions or executing queries. To improve scalability and make the network more efficient, there is a need to research multiple strategies. Primarily efforts should focus on

RQ #4: Can we optimize the gossip layer to prevent nodes from forwarding redundant messages to neighbors?

RQ #5: Secondly, what strategies are needed to ensure nodes route and process transactions only in predefined sections (shards) of the network?

At the moment, this use of resources is not optimized and puts a limit on the network's TPS. Perhaps, the key to resolving the bottleneck lies in the answers to the questions mentioned above.

5.3. Reputation system based on node behavior

Coordicide requires global identities for nodes to implement services such as FPC or CA. The network needs an anti-Sybil mechanism such as node reputations to discourage forging identities. To build this mechanism, there is a need to analyze in detail queries such as:

RQ #6: Will a node reputation system be efficient under both an economic and a game-theoretical perspective?

RQ #7: How are potential aspects that may affect the node reputation gauged, such as the number of tokens transferred, time the node is active, and active or passive participation in the network activity such as voting?

A thorough analysis of these aspects could be investigated to improve the reputation system for nodes.

5.4. Consensus mechanism - CA based consensus

In the Coordicide proposal, a consensus-based voting layer is introduced to resolve conflicts without need for a Coordinator. A possible way for a node to vote is through CA, using the "majority rule": nodes adopt the opinion held by a majority of their neighbors. CA-based voting is vulnerable to Sybil attacks, so further investigation is needed. Primary challenges to the CA-based voting are:

RQ #8: What could be implications for understanding the convergence behavior of CA using the majority rule on random graphs?

RQ #9: Could there be alternatives for increasing the Byzantine resistance of the CA?

RQ #10: Whether it is feasible to include reputation-based approaches or mechanisms that prevent nodes from "lying" about their opinion?

5.5. Theoretical properties of query-based voting schemes

For conflict resolution in a Coordinator-less environment, nodes adopt voting-based strategies. Apart from CA, another way for nodes to vote is using the FPC. Both methods need further investigation of particular topics such as:

RQ #11: What are the theoretical and numerical results on the safety and efficiency of FPC protocol for an optimal implementation.

RQ #12: What is the robustness towards variations of the network topologies, Byzantine resistance, effective implementation of reputation-based systems, and efficient use of decentralized random number generators.

5.6. Efficient algorithms for timestamping of transactions

Although the current version of transactions in IOTA has several fields for timestamps, the core protocol does not rely on timestamps for confirming or filtering transactions. The existence of credible timestamps has several critical advantages for the protocol: E.g., by comparing timestamps, it may be possible to define a global criterion when a transaction becomes "too old" and can be safely removed in a snapshot. They also enable the IOTA protocol to establish a fully ordered Tangle, which is necessary for smart contracts.

The credible timestamps would make it feasible to reach a consensus on the global ordering of the Tangle through timestamps, i.e., declared and signed values attached to each transaction. However, such a schemes success would depend on:

RQ #13: Whether, as a fundamental requirement, such an ordering would be possible with a low network overhead (e.g., low number of votes or even require no voting system)?

RQ #14: Furthermore, would this mechanism be robust to attacks that target the consensus or that aim to increase the number of voting rounds in FPC or CA?

5.7. Scaling through trustless partial Tangle validation

To make IOTA feasible for IoE, it is fundamental to make the Tangle scalable. The network layer optimization and reputation system help improve scalability. However, these systems cannot help the network exceed the intrinsic physical limitations. Two strategies could be implemented to confirm a more significant number of transactions per second: nodes probabilistically validating only a subset of transactions that they receive and each transaction carrying a shard marker to partition the database. With such processes, additional issues could arise, such as:

RQ #15: Previous experiences in sharding DLTs have been complicated, as was seen with Ethereum. What could be a suitable strategy to achieve sharding for IOTA?

RQ #16: How can inter-shard communication be minimized to reduce the network traffic?

5.8. Securing the IOTA protocol

To make sure that the Coordicide solution is resistant to attacks, there is a need to theorize and simulate the system's behavior. This could be achieved by developing new attacking scenarios that could use artificial intelligence, analyzing the cost and feasibility of the proposed attacks, and proposing new security improvements to the protocol.

Just as for Bitcoin, it may happen for IOTA too that a few unlawful clients use the cover of anonymity [75, 14, 33, 85]. It was estimated that in 2017, BTCs worth $770 million were exchanged for unlawful exercises [42], a fourth of bitcoin clients were noxious, and 46% of all bitcoin action was illicit [29]. Critics might then argue that IOTA could become anti-social as it could create obstacles for law enforcement to follow dubious exchanges because of the anonymity and security [72, 30, 56]. With Bitcoin's outstanding growth in transactions during 2012-2016 same phenomena was observed, where clients viz., mixing services [15], betting destinations, exchanging trades, autonomous mining enterprises [32], Ponzi plans, illegal tax avoiders, cheats [52], misappropriators, and blackmailers [73, 55] utilized the cover of secrecy afforded by Bitcoin to misdirect the review trail. To maintain a clean crypto-currency image, IOTA Foundation would need to identify when its crypto-currency is being put to illegal uses. This leads us to the following questions:

RQ #17: Can there be an automated system to detect malicious users on the IOTA network?

RQ #18: A panacea may be offered by AI for following and investigating unlawful clients or exchanges. Existing research on distinguishing criminal operations heavily favors AI in the Bitcoin sphere. Hence, can an AI-based solution be equally fruitful for the IOTA network?

Some topics that have used AI are deanonymizing elements [37, 103, 79, 35], recognizing botnets [53], unlawful exchanges [42], distinguishing dubious bitcoin clients [96, 97, 88, 98, 33, 85] (extortionists [60], ponzi tricks [7], darknet markets [36], ransomwares [54], human dealers [71], frauds [46, 45]), recognizing tax evasion [32, 99, 31], distinguishing blending administrations [51], recognizing bitcoin trades [44], distinguishing illicit exchanges [59, 13], distinguishing bitcoin wallets [1] and bitcoin miners [102]. Even IOTA may encourage AI solutions for recognizing illegal activity in its DLT.

RQ #19: AI solutions require voluminous data, which makes feature engineering or extraction possible. As IOTA DLT is available in digital form, voluminous data is available for feature engineering to drive AI solutions. Therefore, how can AI balance between privacy and yet identify rogue transactions?

5.9. Data sharding

Sharding involves dividing the tangle into multiple independent tangles, each with its own set of nodes to process transactions and manage communication [77]. The multiple DLTs shall interact with each other if a query needs transaction data to be present in more than one tangle. Such a mechanism may be needed to ensure IOTA can reach the goal of "unlimited" TPS. Sharding had been previously proposed to improve the scalability of the Ethereum blockchain [48] and so the same principles are valid even for IOTA.

Technological problems that IOTA DLT may face are to ensure the security of the shards. Hence, we raise the following questions:

RQ #20: Security of shards may be compromised as a subset of the nodes shall secure each shard. In this case, how can the network be secured?

RQ #21: Consensus would be required on the number of shards due to running multiple tangles in parallel. Hence, can a suitable protocol be defined to resolve the issue and decide the number of shards?

RQ #22: Additionally, to be future-proof, a mechanism would be needed to split the network into more shards than the current throughput requires. To maintain connectivity between shards, another "Coordinator" shard would be needed with multiple shards. This Coordinator shard may get overloaded, so an efficient alternate mechanism would be needed to coordinate amongst shards. Therefore, what can be a suitable procedure to reduce the bottlenecks of the Coordinator?

RQ #23: Nodes in the tangle should be allowed to switch over to different shards without incurring high overheads. Can there be an efficient method for such a switch over?

RQ #24: Finally, What security mechanism must be developed to prevent double-spending without high overheads?

Any efficient sharding mechanism would be required to resolve the challenges mentioned above.

5.10. Miscellaneous

Along with the open RQ identified by us, we have compiled suggestions for improvement in the IOTA core protocol proposed by the IOTA users community on the official discord server.

RQ #25: Can a node issue transactions that approve conflicting transactions? What is the likelihood of success of such an attack? What is the feasibility of such an attack in terms of resources?

RQ #26: What is the resource and energy consumption for PoW computation by IoT devices?

RQ #27: What is the resource and energy consumption for performing a double-spending attack on the Tangle?

RQ #28: Is it possible to delay or prevent a transaction from getting approved?

RQ #29: What is the feasibility of a double-spending attack in terms of computing power?

RQ #30: Is it possible to analyze IOTA's behavior using simulations to verify formal work and calculations?

RQ #31: Can IOTA be used as a permissioned blockchain where data/transactions can be accessed by specific participants only? The research department at IOTA agreed to discuss the feasibility of such networks as currently no such facility was being provided [81].

RQ #32: To prove that another transaction indirectly referenced a transaction without having to provide the complete chain of actual transactions between the two [47].

6. Conclusion

There is a shift towards universal connectivity or IoE. This shift has created multiple large-scale and distributed applications and services. However, the service provider protects each service or application, creating an Intranets of Things, which hinders interoperability. Such Intranets of Things creates closed islands of services and obstructs the notion of a connected world. Therefore, it is essential to provide a decentralized trust technology that allows to trade data, manage access, and track responsibilities between various IoE stakeholders.

IOTA Foundation proposed "Tangle" - A DAG-based DLT for maintaining trust in an insecure environment. Tangle allows clients to deploy IoT devices and receive micropayments or share data. All services provided by Tangle are feeless and free. Tangle has found applications in Industrial IoT, Smart Cities, and Autonomous cars, among others. On the other side, the protocol stack is rapidly evolving as Tangle, and its libraries are in infancy.

In February 2020, a large-scale attack was launched against "Trinity wallet" and led to a sizeable loss for stakeholders. So the current study aims to create awareness of the state of the art in IOTA. It analyzes severe and mild security issues, their impact, and their feasibility. The paper also gives solutions in the literature against reported and unreported attacks and discusses open research issues to provide investigators with a direction. The paper has identified 23 different security vulnerabilities in IOTA 1.0, with four yet to be resolved. Further, we have also identified five and eleven attacks and resolutions in IOTA 1.5 and IOTA 2.0, respectively. A total of 32 RQs are provided across ten aspects of IOTA to motivate research and propose a secure IOTA protocol for industry and society. We hope that the current study shall instigate interest amongst the scientific community to focus on IOTA.

References

[1] Aiolli, F., Conti, M., Gangwal, A., Polato, M., 2019. Mind your wallet's privacy: Identifying bitcoin wallet apps and user's actions through network traffic analysis. Proceedings of the 34th ACM Symposium on Applied Computing , 1–16.

[2] Attias, V., Bramas, Q., 2019. How to choose its parents in the tangle. International Conference on Networked Systems 1, 275–280.

[3] Attias, V., Vigneri, L., Dimitrov, V., 2020. Preventing denial of service attacks in iot networks through verifiable delay functions. arXiv e-prints 2006.01977, 1–6.

[4] Attias, V., Vigneri, L., Dimitrov, V., 2021. Implementation study of two verifiable delay functions. International Conference on Blockchain Economics, Security and Protocols .

[5] Bachmann, S., 2019. Analysis of the tangle in the iot domain. MSc degree in Department of Informatics, University of Zurich 1, 1–43.

[6] Bahar Farahani, Farshad Firouzi, M.L., 2021. The convergence of iot and distributed ledger technologies (dlt): Opportunities, challenges, and solutions. Journal of Network and Computer Applications 177.

[7] Bartoletti, M., Pes, B., Serusi, S., 2018. Data mining for detecting bitcoin ponzi schemes. Crypto Valley Conference on Blockchain Technology , 75–84.

[8] Bartolomeu, P., Vieira, E., Ferreira, J., 2018. Iota feasibility and perspectives for enabling vehicular applications. IEEE Globecom Workshops 1, 1–7.

[9] Bera, B., Das, A., Obaidat, M., Vijayakumar, P., Hsiao, K., Park, Y., 2021. Ai-enabled blockchain-based access control for malicious attacks detection and mitigation in ioe. IEEE Consumer Electronics Magazine 10, 82–92. doi:10.1109/MCE.2020.3040541.

[10] Bernabe, J., Canovas, J., Hernandez, R., Jose, L., Moreno, R., Skarmeta, A., 2019. Privacy-preserving solutions for blockchain: review and challenges. IEEE Access 7, 164908–164940.

[11] Bernstein, D., 2020 (accessed March 3, 2022). Ed25519: high-speed high-security signatures. URL: https://ed25519.cr.yp.to/.

[12] Blockchain, the General Data Protection Regulation, 2019 (accessed September 3, 2021). URL: https://www.europarl.europa.eu/RegData/etudes/STUD/2019/634445/EPRS_STU(2019)634445_EN.pdf.

[13] Bogner, A., 2017. Seeing is understanding: anomaly detection in blockchains with visualized features. Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers , 5–8.

[14] Bohannon, J., 2016. The bitcoin busts. American Association for the Advancement of Science 351, 1144–1146.

[15] Bonneau, J., Narayanan, A., Miller, A., Clark, J., Kroll, J., Felten, E., 2014. Mixcoin: Anonymity for bitcoin with accountable mixes, in: International Conference on Financial Cryptography and Data Security, Springer. pp. 486–504.

[16] Bramas, Q., 2018. The stability and the security of the tangle. IOTA Foundation Technical Report 1, 1–12.

[17] Buchmann, J., Dahmen, E., Ereth, S., Hülsing, A., Rückert, M., 2013. On the security of the winternitz one-time signature scheme. International Journal of Applied Cryptography 3, 84–96.

[18] Buffy, B., 2020 (accessed September 3, 2021)a. If exaggerates chrysalis tps by 4x. URL: https://facemrook.github.io/chrysalis1-tps.

[19] Buffy, B., 2020 (accessed September 3, 2021)b. White flag is a mistake. URL: https://facemrook.github.io/chrysalis1-whiteflag.html.

[20] Buffy, B., 2020 (accessed September 3, 2021)c. Zero-value transactions: Fun but worthless. URL: https://facemrook.github.io/zero-value#fn:conflicts.

[21] Cai, D., 2019. A parasite chain attack in iota. University of Twente 1, 1–112.

[22] Capossele, A., Mueller, S., Penzkofer, A., 2019. Robustness and efficiency of leaderless probabilistic consensus protocols within byzantine infrastructures. arXiv preprint 1911.08787, 1–21.

[23] Choo, K., Yan, Z., Meng, W., 2020. Blockchain in industrial iot applications: Security and privacy advances, challenges and opportunities. IEEE Transactions on Industrial Informatics 16, 4119–4121.

[24] Conti, M., Kumar, E., Lal, C., Ruj, S., 2018. A survey on security and privacy issues of bitcoin. IEEE Communications Surveys & Tutorials 20, 3416–3452.

[25] Cullen, A., Ferraro, P., King, C., Shorten, R., 2019. Distributed ledger technology for smart mobility: Variable delay models. IEEE Conference on Decision and Control 1, 8447–8452.

[26] Cullen, A., Ferraro, P., Sanders, W., Vigneri, L., Shorten, R., 2020. On congestion control for distributed ledgers in adversarial iot networks. arXiv e-prints 2005.07778, 1–7.

[27] Ding, D., Han, Q.L., Xiang, Y., Ge, X., Zhang, X.M., 2018. A survey on security control and attack detection for industrial cyber-physical systems. Neurocomputing 275, 1674–1683. URL: https://www.sciencedirect.com/science/article/pii/S0925231217316351, doi:https://doi.org/10.1016/j.neucom.2017.10.009.

[28] Ferraro, P., King, C., Shorten, R., 2020. On the stability of unverified transactions in a dag-based distributed ledger. IEEE Transactions on Automatic Control 65, 3772–3783.

[29] Foley, S., Karlsen, J., Putniņš, T., 2019. Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? The Review of Financial Studies 32, 1798–1853.

[30] Ghosh, A., Gupta, S., Dua, A., Kumar, N., 2020. Security of cryptocurrencies in blockchain technology: State-of-art, challenges and future prospects. Journal of Network and Computer Applications .

[31] Harlev, M., Sunyin, H., Langenheldt, K., Mukkamala, R., Vatrapu, R., 2018. Breaking bad: De-anonymising entity types on the bitcoin blockchain using supervised machine learning. Proceedings of the 51st Hawaii International Conference on System Sciences , 1–14.

[32] Hu, Y., Seneviratne, S., Thilakarathna, K., Fukuda, K., Seneviratne, A., 2019. Characterizing and detecting money laundering activities on the bitcoin network. arXiv 1912.12060, 1–16.

[33] Irwin, A., Turner, A., 2018. Illicit bitcoin transactions: challenges in getting to the who, what, when and where. Journal of money laundering control .

[34] Janečko, T., Zelinka, I., 2018. Impact of security aspects at the iota protocol. International Conference on Intelligent Information Technologies for Industry 1, 41–48.

[35] Jourdan, M., Blandin, S., Wynter, L., Deshpande, P., 2018. Characterizing entities in the bitcoin blockchain, in: IEEE International Conference on Data Mining Workshops, IEEE. pp. 55–62.

[36] Kanemura, K., Toyoda, K., Ohtsuki, T., 2019. Identification of darknet markets' bitcoin addresses by voting per-address classification results. IEEE International Conference on Blockchain and Cryptocurrency , 154–158.

[37] Kumar, A., Abhishek, K., Nerurkar, P., Khosravi, M., Ghalib, M., Shankar, A., 2021. Big data analytics to identify illegal activities on bitcoin blockchain for iomt. Personal and Ubiquitous Computing , 1–12.

[38] Kusmierz, B., 2017. The first glance at the simulation of the tangle: discrete model. IOTA Foundation Technical Report 1, 1–10.

[39] Kusmierz, B., Gal, A., 2018. Probability of being left behind and probability of becoming permanent tip in the tangle v0. 2. IOTA Foundation Technical Report 1, 1–9.

[40] Kusmierz, B., Sanders, W., Penzkofer, A., Capossele, A., Gal, A., 2019. Properties of the tangle for uniform random and random walk tip selection. IEEE International Conference on Blockchain 1, 228–236.

[41] Kusmierz, B., Staupe, P., Gal, A., 2018. Extracting tangle properties in continuous time via large-scale simulations. IOTA Foundation Technical Report 1, 1–10.

[42] Lee, C., Maharjan, S., Ko, K., Hong, J., 2020. Toward detecting illegal transactions on bitcoin using machine-learning methods, in: Zheng, Z., Dai, H., Tang, M., Chen, X. (Eds.), Blockchain and Trustworthy Systems, Springer Singapore, Singapore. pp. 520–533.

[43] Li, Y., Cao, B., Peng, M., Zhang, L., Zhang, L., Feng, D., Yu, J., 2020. Direct acyclic graph-based ledger for internet of things: Performance and security analysis. IEEE Transactions on Networking , 1–12.

[44] Liang, J., Li, L., Luan, S., Gan, L., Zeng, D., 2019. Bitcoin exchange addresses identification and its application in online drug trading regulation. Pacific Asia Conference on Information systems , 1–17.

[45] Monamo, P., Marivate, V., 2016. A multifaceted approach to bitcoin fraud detection: Global and local outliers, in: 15th IEEE International Conference on Machine Learning and Applications, IEEE. pp. 188–194.

[46] Monamo, P., Marivate, V., Twala, B., 2016. Unsupervised learning for robust bitcoin fraud detection, in: 2016 Information Security for South Africa (ISSA), IEEE. pp. 129–134.

[47] Moog, H., 2020 (accessed December 26, 2020). Merkle proofs of inclusion. URL: https://iota.cafe/t/merkle-proofs-of-inclusion/248.

[48] Moog, H., 2020 (accessed October 6, 2020). Data sharding. URL: https://medium.com/@hans_94488/scaling-iota-part-1-a-primer-on-sharding-fa1e2cd27ea1.

[49] Moog, H., 2020 (accessed September 3, 2021). Switching to utxo model for balances. URL: https://iota.cafe/t/switching-to-utxo-model.

[50] Moser, L., 2020 (accessed September 3, 2021). Atomic transactions. URL: https://github.com/luca-moser/protocol-rfcs/blob/signed-tx-payload/text/0000-transaction-payload/0000-transaction-payload.md.

[51] Nan, L., Tao, D., 2018. Bitcoin mixing detection using deep autoencoder. IEEE Third International Conference on Data Science in Cyberspace , 280–287.

[52] Nerurkar, P., Bhirud, S., Patel, D., Ludinard, R., Busnel, Y., Kumari, S., 2021a. Supervised learning model for identifying illegal activities in bitcoin. Applied Intelligence 51, 3824–3843.

[53] Nerurkar, P., Busnel, Y., Ludinard, R., Shah, K., Bhirud, S., Patel, D., 2020. Detecting illicit entities in bitcoin using supervised learning of ensemble decision trees, in: Proceedings of the 2020 10th international conference on information communication and management, pp. 25–30.

[54] Nerurkar, P., Patel, D., Busnel, Y., Ludinard, R., Kumari, S., Khan, M., 2021b. Dissecting bitcoin blockchain: Empirical analysis of bitcoin network (2009–2020). Journal of Network and Computer Applications 177, 102940.

[55] Paquet-Clouston, M., Romiti, M., Haslhofer, B., Charvat, T., 2019. Spams meet cryptocurrencies: Sextortion in the bitcoin ecosystem, in: Proceedings of the 1st ACM Conference on Advances in Financial Technologies, pp. 76–88.

[56] Park, S., Im, S., Seol, Y., Paek, J., 2019. Nodes in the bitcoin network: comparative measurement study and survey. IEEE Access 7, 57009–57022.

[57] Penzkofer, A., 2020 (accessed January 4, 2021). Orphanage with restricted urts. URL: https://iota.cafe/t/orphanage-with-restricted-urts/1199.

[58] Penzkofer, A., Kusmierz, B., Capossele, A., Sanders, W., Saa, O., 2020. Parasite chain detection in the iota protocol. arXiv e-prints 2004.13409, 1–16.

[59] Pham, T., Lee, S., 2016. Anomaly detection in bitcoin network using unsupervised learning methods. arXiv 1611.03941, 1–13.

[60] Phetsouvanh, S., Oggier, F., Datta, A., 2018. Egret: Extortion graph exploration techniques in the bitcoin network, in: IEEE International Conference on Data Mining Workshops, pp. 244–251.

[61] Popov, S., 2016. The tangle. IOTA Foundation Technical Report 1, 131–156.

[62] Popov, S., 2018. Local modifiers in the tangle. IOTA Foundation Technical Report 1, 1–9.

[63] Popov, S., 2020 (accessed December 26, 2020). Coordicide grant program. URL: https://coordicide.iota.org/grants.

[64] Popov, S., 2020 (accessed December 30, 2020). Research papers - iota. URL: https://www.iota.org/foundation/research-papers.

[65] Popov, S., 2020 (accessed October 6, 2020)a. Getting started as a node. URL: https://docs.iota.org/docs/.

[66] Popov, S., 2020 (accessed October 6, 2020)b. Trinity attack incident part 1: Summary and next steps. URL: https://blog.iota.org/.

[67] Popov, S., 2020 (accessed September 3, 2021). Getting started as a node. URL: https://legacy.docs.iota.org/docs/getting-started/1.1/running-nodes/running-a-node.

[68] Popov, S., Buchanan, W., 2019. Fast probabilistic consensus within byzantine infrastructures. arXiv preprint 1905.10895, 1–30.

[69] Popov, S., Moog, H., Camargo, D., Capossele, A., Dimitrov, V., Gal, A., Greve, A., Kusmierz, B., Mueller, S., Penzkofer, A., 2020. The coordicide. IOTA Foundation Technical Report 1, 1–42.

[70] Popov, S., Saa, O., Finardi, P., 2019. Equilibria in the tangle. Computers & Industrial Engineering 136, 160–172.

[71] Portnoff, R., Yuxing, D., Doerfler, P., Afroz, S., McCoy, D., 2017. Backpage and bitcoin: Uncovering human traffickers. KDD '17 , 1–12.

[72] Rahouti, M., Xiong, K., Ghani, N., 2018. Bitcoin concepts, threats, and machine-learning security solutions. IEEE Access 6, 67189–67205.

[73] Reyes-Macedo, V., Salinas-Rosales, M., Garcia, G., 2019. A method for blockchain transactions analysis. IEEE Latin America Transactions 17, 1080–1087.

[74] Rogozinski, G., Welz, W., 2020 (accessed December 26, 2020). Coordinator improvements. URL: https://iota.cafe/t/coordinator-improvements/310.

[75] Sabry, F., Labda, W., Erbad, A., Jawaheri, H., Malluhi, Q., 2019. Anonymity and privacy in bitcoin escrow trades, in: Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society, pp. 211–220.

[76] Sanders, W., 2020 (accessed December 26, 2020). A different proposal for a tsa in chrysalis. URL: https://iota.cafe/t/a-different-proposal-for-a-tsa.

[77] Sanders, W., 2020 (accessed October 6, 2020). Data sharding. URL: https://iota.cafe/t/data-sharding/1188.

[78] Schwerin, S., 2018. Blockchain and privacy protection in the case of the european general data protection regulation (gdpr): a delphi study. The Journal of the British Blockchain Association 1, 3554.

[79] Shao, W., Li, H., Chen, M., Jia, C., Liu, C., Wang, Z., 2018. Identifying bitcoin users using deep neural network, in: Algorithms and Architectures for Parallel Processing, Springer International Publishing, Cham. pp. 178–192.

[80] Staupe, P., 2017. Quasi-analytic parasite chain absorption probabilities in the tangle. IOTA Foundation Technical Report 20, 15–18.

[81] Stähle, J., 2020 (accessed December 26, 2020). Iota as permissioned network? URL: https://iota.cafe/t/iota-as-permissioned-network/1180.

[82] Tennant, L., 2017. Improving the anonymity of the iota cryptocurrency. IOTA Foundation Technical Report 1, 1–27.

[83] Thakare, A., Kim, Y.G., 2021. Secure and efficient authentication scheme in iot environments. Applied Sciences 11, 1260.

[84] Tran, N.K., Ali Babar, M., Boan, J., 2021. Integrating blockchain and internet of things systems: A systematic review on objectives and designs. Journal of Network and Computer Applications 173, 102844.

[85] Turner, A., Irwin, A., Maitland, S., 2018. Bitcoin transactions: a digital discovery of illicit activity on the blockchain. Journal of Financial Crime .

[86] Vigneri, L., Welz, W., 2020. On the fairness of distributed ledger technologies for the internet of things, in: IEEE International Conference on Blockchain and Cryptocurrency, pp. 1–3.

[87] Wang, D., Zhao, J., Wang, Y., 2020. A survey on privacy protection of blockchain: the technology and application. IEEE Access 8, 108766–108781.

[88] Weber, M., Domeniconi, G., Chen, J., Weidele, D., Bellei, C., Robinson, T., Leiserson, C., 2019. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. arXiv 1908.02591, 1–13.

[89] Welz, W., 2019 (accessed September 3, 2021). White flag conflict spamming. URL: https://iota.cafe/t/conflict-white-flag-mitigate.

[90] Welz, W., 2020 (accessed December 26, 2020). Conflict spamming attack. URL: https://iota.cafe/t/conflict-spamming-attack/232.

[91] Welz, W., 2020 (accessed October 6, 2020). Coordinator improvements for mitigating blowball attacks. URL: https://iota.cafe/t/coordinator-improvements/310.

[92] Welz, W., 2020 (accessed September 3, 2021). Ed25519 signature scheme. URL: https://github.com/iotaledger/protocol-rfcs/blob/ee07797acb5940b7dbb5c3411b184ccdc6afdbb1/text/0000-ed25519-signature-scheme/0000-ed25519-signature-scheme.md.

[93] Welz, W., 2022 (accessed March 3, 2022). Sea of signature schemes. URL: https://www.helloiota.com/articles/signature-schemes.

[94] Welz, W., Rogozinski, G., 2020 (accessed December 26, 2020). Hetfield solution for conflict spamming: Ignore everything conflicting. URL: https://iota.cafe/t/hetfield-solution.

[95] Welz, W., Thompson, C., Andrea, V., Rogozinkski, G., 2020 (accessed September 3, 2021). Weighted uniform random tip selection. URL: https://github.com/iotaledger/protocol-rfcs.

[96] Wu, Y., Luo, A., Xu, D., 2019. Identifying suspicious addresses in bitcoin thefts. Digital Investigation 31, 200–212.

[97] Wu, Y., Tao, F., Liu, L., Gu, J., Panneerselvam, J., Zhu, R., Shahzad, M.N., 2020. A bitcoin transaction network analytic method for future blockchain forensic investigation. IEEE Transactions on Network Science and Engineering , 1–1.

[98] Yang, L., Dong, X., Xing, S., Zheng, J., Gu, X., Song, X., 2019. An abnormal transaction detection mechanim on bitcoin, in: International Conference on Networking and Network Applications, IEEE. pp. 452–457.

[99] Yin, H., Vatrapu, R., 2017. A first estimation of the proportion of cybercriminal entities in the bitcoin ecosystem using supervised machine learning, in: IEEE International Conference on Big Data (Big Data), IEEE. pp. 3690–3699.

[100] Zander, M., 2018. A multi-agent simulation framework and analysis of the iota tangle. MSc degree in Computing Science of Imperial College London 1, 1–88.

[101] Zander, M., Waite, T., Harz, D., 2019. Dagsim: Simulation of dag-based distributed ledger protocols. ACM SIGMETRICS Performance Evaluation Review 46, 118–121.

[102] Zayuelas, M., 2019. Detection of Bitcoin miners from network measurements. B.S. thesis. Universitat Politècnica de Catalunya.

[103] Zola, F., Eguimendia, M., Bruse, J., Urrutia, R., 2019. Cascading machine learning to attack bitcoin anonymity, in: IEEE International Conference on Blockchain, IEEE. pp. 10–17.