2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering

Evaluating DAG-based Blockchains for IoT

Liangrong Zhao
School of Electronic and Information Engineering
Beijing Jiaotong University
Beijing, China
liangrongzhao@bjtu.edu.cn

Jiangshan Yu
Faculty of Information Technology
Monash University
Melbourne, Australia
jiangshan.yu@monash.edu

***Abstract*—Blockchain has been considered as the enabler to provide automated and trust-less Internet of things (IoT) services. However, legacy consensus mechanisms applied in most of the blockchain networks pose obstacles for deployment in IoT in respect of decentralization and scalability. The emerging Directed Acyclic Graph (DAG)-based models aim at delivering the benefits of blockchain without making compromise on performance. This paper gives a critical analysis on the two representative DAG-based models, IOTA and Hashgraph. In particular, we provide an insightful overview of the two blockchains, with their features and functionalities. We define desired criteria for evaluating these two blockchains, provide an in-depth analysis on the two systems, and identify open challenges IOTA and Hashgraph are facing for their application in IoT.**

***Index Terms*—blockchain, IOTA, DAG, Hashgraph, IoT**

I. Introduction

The Internet of things (IoT) nowadays is in continuous expansion with ever-growing popularity [1]. Adoption of IoT based technology will have a huge influence on various sectors of our lives by generating, processing, and exchanging vast amount of security and safety-critical data as well as privacy-sensitive information, which on the other hand makes IoT devices appealing targets of various cyberattacks [2]. With more and more deployments of potentially large scale IoT systems, ensuring security, scalability, and confidentiality of the IoT networks becomes increasingly critical and key benchmarks for IoT ecosystems [3]. Current centralized security frameworks make it difficult for IoT to expand in scale due to the many-to-one nature of traffic, whose complexity is exponentially increasing as the network scales up [4].

Blockchain has been considered to be a solution to tackle the issue of security and privacy in IoT with concepts of decentralization and anonymity [5]. There are efforts to utilize blockchain to facilitate peer-to-peer communication among IoT devices to avoid a centralized network where a massive number of devices can be manipulated to initiate a DDoS attack if the center node of this network is malicious or infected by malware. Cryptocurrencies, which are probably the most famous applications of blockchain, are presumably a referable case for IoT because users in cryptocurrencies make transactions in similar way devices are supposed to exchange information in IoT.

Among all the consensus mechanisms applied in various cryptocurrencies, Proof of work (PoW) is probably the most popular and representative one. PoW-based platforms assume honest users are in control of the majority of the computing power and users are motivated to join and maintain the network for the reward of finding a new block. The possibility of a node finding a new block is determined by the proportion of this node's computing power in the entire network. Consequently, the overall computing power possessed by the network usually increases rapidly due to the users' strong incentive to expand their computing power in order to get more rewards. A network with stronger computing power is supposed to be safer because the cost to initiate an attack will be accordingly higher. A typical example of PoW-based cryptocurrency is the famous Bitcoin [6].

Generally, current IoT systems deployed with blockchain use a server-client model where identification, authentication and communication are realized through servers with high computation and storage resources [7]. Such arrangement can avoid all IoT devices being trapped in heavy computation tasks and lower the threshold for IoT devices to participate in blockchain networks.

Those who think this is an ideal usage failed to see there are plenty of scenarios where most of the devices are simply-built and incapable of strong computing. If the legacy consensus algorithms in cryptocurrencies are directly applied in those IoT systems, several devices with stronger computing power or storage resources will control the network, which will consequently become a de facto centralized system.

To tackle this issue, [8] proposed a novel PoW-free blockchain structure that is similar to a permissioned blockchain with each private chain representing IoT devices group owned by the same user, like a smart home. Lots of approaches are based on cutting down the number of devices to a controllable scale, mature mechanisms used in public is usually unavailable or poorly performing. Generally, the majority of solutions to tackle it are involved with a "center node", which is in the role of either a gateway to communicate with the Internet or a scheduler in charge of authentication. However, such instantiation of blockchain can only bring the blockchain's benefits within the limited IoT intranet. When communicating with the real Internet, they still need the outlet of a centralized gateway, which basically loses advantages of peer-to-peer communication.

An exponentially growing number of IoT devices in the near future are expected to generate trillions of transactions every day [9]. Consequently, weak scalability and long transaction

2324-9013/19/$31.00 ©2019 IEEE
DOI 10.1109/TrustCom/BigDataSE.2019.00074

time in legacy cryptocurrency systems like Bitcoin have to be improved in IoT scenarios.

Various approaches utilizing Directed Acyclic Graph (DAG) have emerged to make blockchain faster and more scalable for micropayments. Theoretically, DAG is not blockchain as the valid transactions are not arranged in the shape of a chain. Instead, transactions in DAG are allowed to fork as often as they please and consequently in the shape of a tree. However, we still consider DAG as a special version of blockchain and compare it with blockchain because they are aimed to solve the same issue and have similar applications. DAG can bring the features that legacy blockchain is scrambling to deliver, particularly the ones that are crucial in IoT scenarios, including better scalability and shorter transaction time. Hence, DAG is considered to be an emerging game-changer for blockchain-based IoT ecosystems.

Among all the DAG-based models, two typical ones are IOTA [10] and Swirlds hashgraph consensus (Hashgraph) [11], loosely representing classic cryptocurrency platform and database-replication protocol. They are selected for comparison to outline the available solutions the DAG can provide to tackle the issues the blockchain fails to handle in IoT.

In this paper, we summarize the key features of IOTA and Hashgraph. In addition, we analyze the functionality of these two DAG-based models based on the criteria desired by blockchain for IoT scenarios. The rest of the paper is organized as follows: Section II and Section III outline the main features of IOTA and Hashgraph respectively. Section IV introduces the criteria to evaluate two models' performance in IoT. Section V gives the analyzation of the models' capacities in respect of the aforementioned criteria.

II. IOTA

In this section, we will cover the general characteristics of IOTA and the main operation procedures. The data structure at the heart of IOTA's DAG-based consensus mechanism is called Tangle, in which each vertex represents a transaction and an edge is a transaction's approval to a previous one. It's worth noticing that unlike other DAG-based cryptocurrencies where a vertex is a block [12], Tangle adopts a block-free design, which is more convenient for instant micropayments.

A. Operation Procedures

Every incoming transaction in IOTA is supposed to approve two unapproved transactions, which are called tips in Tangle. As depicted in Fig. 1, arrows are approvals from a transaction to another one and dark gray vertices are unapproved tips while white ones are approved transactions. Approving a tip in this figure means adding an arrow pointing to that tip and a tip can be approved by lots of subsequent transactions. When an incoming transaction successfully approves two tips and becomes a new tip, it is possible that incoming transactions still approve the previous tips that have already been approved as shown in the figure. The reason is that network delay makes incoming transactions fail to discover new generated tips and those delayed nodes still consider those newly approved tips as unapproved ones. As a consequence, approving an already approved transaction is allowed but not encouraged, which is an important rule in IOTA.

Approving a transaction means to verify whether the balances in this transaction are negative or this transaction is conflicting with the previously approved transactions. By approving two tips, this transaction becomes a tip and waits for approvals from newly incoming transactions. It is noteworthy that selecting and approving tips can be performed independently by different users, which means a user does not have to wait until other users have done their job. Different transactions can be appended to Tangle simultaneously. Before introducing some more concrete features, it is necessary to clarify that when transaction A is approved by B and B is approved by C, we can say A is directly approved by B and indirectly approved by C. Roughly, if a transaction is approved (directly and indirectly) sufficiently, it will be part of Tangle history and barely reversible. More details about it will be given in later sections.

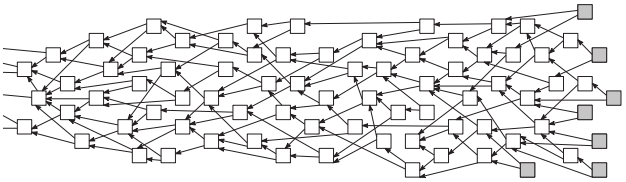

Fig. 1. Tangle, DAG-based data structure of IOTA [10]

B. Tip Selection

a) Unweighted Random Walk: The tip-selecting strategy is significantly important in Tangle as the stability of system is closely relevant to it. In order to not orphan any tips, an algorithm called unweighted random walk is introduced to walk randomly from the very first transaction, the genesis transaction, towards tips. Theoretically, every tip is evenly probable to be chosen and approved. This works well until some lazy user only approves old transactions other than new tips because it does not bother keeping up to date with the latest state of Tangle. In unweighted random walk, this lazy user's transaction will not be ignored or penalized and it is even encouraged in some particular cases. It's obvious that this behavior should be penalized or prohibited, otherwise less and less users will follow the rule to approve the tips and the network will cease to work.

b) Weighted Random Walk: With that in mind, weighted random walk is a much better solution. The main idea of weighted random walk is to bias the tips that approve latest transactions instead of old ones so that transactions issued by lazy users are less likely to be approved and gradually be orphaned. A parameter called *cumulative weight* is used to indicate how bias a tip is. The cumulative weight of each tip is the sum of the direct and indirect approvers plus one. Loosely speaking, selecting tips in this algorithm is to follow the path of vertexes with the heaviest cumulative weight, which is

similar with the longest chain selection in Bitcoin [6]. The lazy user's tip in the previous example will have a much smaller cumulative weight than normal tips because a lazy user's branch, which is always attached to an old transaction, is always shorter and has less indirect approvers than normal ones. Therefore, lazy users will be less and less likely to be approved because the Tangle graph is growing and the gap between lazy tips and the normal tips is growing. It's worth noticing that a relatively low cumulative weight makes the tip less likely to be approved by incoming transactions but not impossible. More precisely, a parameter α (from 0 to 1) is set to determine how strong cumulative weight is in tip selection. The reason α is not supposed to be set to 1 to make tip selection entirely depends on cumulative weight is that sometimes honest users accidentally approve an old transaction due to delay, if α is set to 1, these honest tips will be orphaned for good. Therefore, some randomness is necessary and finding an optimal α is of great importance in Tangle.

If a transaction receives additional approvals, its cumulative weight grows and it will be more difficult to be overthrown to initiate double-spending attack, which will be covered in Section IV.

III. Hashgraph

A. Functionality

Hashgraph is a consensus algorithm for replicated state machines with guaranteed Byzantine fault tolerance (BFT). Term *Byzantine* here is used in the strong sense of its original definition: up to less than 1/3 of the members can be attackers who are able to collude with each other to delay messages between honest members without bounds [13]. According FLP theorem [14], no deterministic Byzantine system can be completely asynchronous with unlimited message delays and still guarantee BFT consensus. Different from the practical Byzantine fault tolerance to implement a deterministic consensus with weak synchrony [15], Hashgraph is a completely asynchronous nondeterministic consensus algorithm capable of achieving Byzantine agreement with probability one without leaders or round robin that are required in other BFT solutions. To achieve that, Hashgraph initiates two innovative features: *gossip about gossip* and *virtual voting*. Before we outline gossip about gossip, we have to introduce a data structure that bears the same name with the algorithm, hashgraph (not confused with Hashgraph). What makes hashgraph unique is that it records not only the content of the message but also whom this message is from in what order.

B. Gossip about Gossip

Gossip about gossip is the protocol used by Hashgraph to spread messages through the network. What makes it different from the legacy gossip protocol is that nodes in Hashgraph not only gossip about the event (similar to block in Bitcoin), but also the history of gossip itself. By doing so, each node will maintain its own hashgraph as shown in Fig. 2. Each event they gossip includes several transactions in order and two hash pointers pointing to this event's two parents, which we are going to introduce.

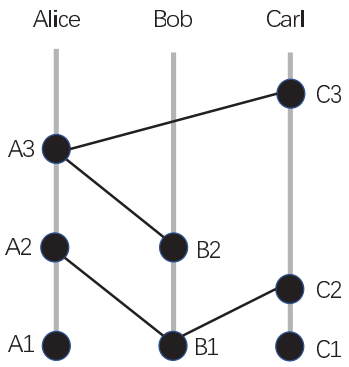

Fig. 2. An example of data structure maintained by each node to show Gossip history as a directed graph

Fig. 2 is an example to show how gossip about gossip works: Bob gossips his event to Alice, including all the ordered transactions in the event and whom he heard from. After hearing Bob's gossip, Alice merges Bob's event (B1) with her own event (A1) and it will become Alice's new event (A2) in her hashgraph. Alice's previous event will be her current event's self-parent and Bob's event will be other-parent. Similarly, Carl merges his event (C1) with Bob's and gets a new event (C2) after gossip about gossip from Bob. After a period of time, Bob gossips his new event (B2), including some new transactions he received, to Alice who might also have some new transactions after the last gossip. Alice's hashgraph is updated with the latest events from both Bob and Alice before Alice gossips her event to Carl. Fig. 2 is the hashgraph maintained by Carl. Through gossip about gossip, Carl can know all the events and the sources of those events. If Carl later gossips to other users, then the rest of the network is supposed to know this hashgraph.

The self-parent of the self-parent is self-ancestor. In Fig. 2, the red is the self-parent of yellow and dark green is its other-parent. The red is the self-ancestor of orange. It needs to be mentioned that one of the two events from the same user has to be self-ancestor (parent and ancestor are not strictly distinguished in this case), which is a critical notion to tackle malicious attacks that we will cover in the following section.

Even though synchrony is not required in Hashgraph, transactions are expected to be gossiped throughout the network fairly quickly in most of the cases, thanks to the convergency of gossip protocol [16]. Since Hashgraph is an asynchronous algorithm, it cannot be guaranteed that all the nodes will have the same state of hashgraph at any given time. However, the hashgraph maintained by different nodes will be consistent, which means they will eventually reach consensus about a version of Hashgraph of an exact moment in the past. This paves the way for virtual voting.

C. Virtual Voting

Virtual voting is to reach consensus among nodes about the order of events. The most distinguished feature of virtual voting is that nodes can get election results through local calculation, which allows them to get free of the bandwidth-consuming "message storm" to exchange everyone's vote and acknowledgment. As mentioned before, each node maintains a hashgraph to record events in the network and keeps it up to date through gossip about gossip. Consequently, a node can be confident to anticipate other nodes' votes about the transaction's order based on the history of gossip in hashgraph. If there is an attacker involved who sends different versions of gossip to different nodes and subsequently causes a fork in its ancestor history, it will be spotted by honest users when they gossip about gossip with each other and the malicious user will be excluded from the later visual voting. We will get into details of it in the following section. Since Hashgraph is asynchronous, nodes do not expect to get the voting result simultaneously but they will eventually have the same voting result considering they maintain the same order of transactions in hashgraph.

IV. Evaluation Criteria

In this section, we will introduce several criteria to evaluate the performance of IOTA and Hashgraph. The evaluation is aimed to explore the feasibility of DAG-based solutions in IoT. For the record, we will not unfairly judge the overall functionality of IOTA and Hashgraph because IOTA is a platform specifically designed for IoT while Hashgraph is not exclusively for IoT. Instead, we evaluate some concrete schemes they employ to tackle the same issue to show their potential in IoT scenarios. With that being said, our criteria cannot represent all the challenges the blockchain-based IoT networks are facing as there are still other factors that can be of great importance, such as authentication or supervision. The benchmarks we pick here are what we suppose to be the most critical for the comparison between IOTA and Hashgraph.

Following are the properties we will compare between IOTA and Hashgraph, as shown in Table I.

- **Network synchrony**: A property shows whether the users are guaranteed to keep up to date with the latest changes in the network.
- **Storage overhead**: Storage capacity required for users to participate in maintaining the network.
- **Finality**: A property to prove that once a transaction is completed, there is no way to revert it.
- **Performance**: The maximum number of transactions per second (tps) the network can handle on average.
- **Tolerated power of an adversary**: The maximum power the network can tolerate from colluded malicious users who intend to initiate attacks.
- **Scalability**: The ability to handle an increasing number of users and transactions without making compromise on functionality or performance.
- **Incentive**: The motivation for users to join and stick to the network and propagate transactions [17]. It also includes the penalty "lazy users" receive when they refuse to follow the general rules or take a passive attitude towards expected duty.

V. Comparison between IOTA and Hashgraph

A. Storage Overhead

Only full nodes with the storage of the entire Tangle network are able to issue transactions, including tip selecting, validating and broadcasting transactions. Light nodes are also allowed in IOTA network without the need to download the data of the entire network. What makes light nodes capable of operating normally in the network is that light nodes must connect to a special kind of full nodes called public nodes who will do the whole transaction-issuing task on behave of light nodes. As convenient and advantageous as light nodes may sound, they have to completely trust the third-party public nodes to perform critical functions that require sensitive information like balances and transaction history. Obviously, there are still a huge number of IOTA users who maintain full nodes to do transactions for the sake of their information security even if they have to download all the transactions from the genesis transactions to the newest tips. In order to lower the threshold for full nodes, IOTA is performing snapshots of IOTA network to prune the database on a more or less regular basis. At the end of 2018, the size of a full node is over 50GB.

Nodes in Hashgraph must be full nodes mostly because they have to store the gossip history in the network to perform virtual voting. However, it doesn't mean the nodes have to store every single transaction in the history from the genesis to the latest event because after each round, the event before that round could be erased as long as the balances of each user are kept. It's not impossible for platforms based on Hashgraph to introduce a concept like light nodes, considering in most of the cryptocurrencies lightweight nodes can be facilitated with most of the functions by letting a full node do their job. However, how to implement a lightweight client for IOTA and Hashgraph, such that resource limited devices can verify transactions without the need of trusting any third party, is still an open challenge.

B. Network Synchrony

Both IOTA and Hashgraph are asynchronous, as they claimed in whitepapers. Their asynchrony is based on the concept that a period time of being out of network synchronization won't make nodes stop functioning. Therefore, synchrony is not necessarily required in both models.

Asynchrony is assumed in IOTA because the nodes do not have to see the same set of transactions all the time. Full nodes in IOTA are set to be connected with 7 neighboring nodes when they access network for the first time. They keep up to date with the newest version of Tangle network by communicating with the neighboring nodes. Asynchrony of IOTA doesn't require nodes to be instantly up to date and doesn't guarantee the data can be delivered to each node eventually. If they approve the old transactions because they fail to get the latest version of Tangle to perform tip selection,

Table I. Comparison between IOTA and Hashgraph

	IOTA	Hashgraph
Network synchrony	Not required	Not required
Storage overhead	Large (full nodes)	Medium
Finality	Yes (with coordinator) No (without coordiantor)	Yes
Performance	Sufficient	Excellent
Tolerated power of an adversary	50% of network's capability	1/3 of voting power
Scalability	Good in users and load	Excellent in load Medium in users
Incentive	Feeless	Fees involved (Hedera)

their transactions won't be invalid because the tip selection algorithm can tolerate situations like this.

Synchrony is usually not necessarily needed for safety in BFT consensus algorithms. Hashgraph is also an asynchronous algorithm where nodes do not have to synchronized for all the events gossiped in the network. Asynchrony in Hashgraph is achieved by the locally performed voting section that doesn't require being at the same step with any other users. Hashgraph doesn't guarantee any message delivery between nodes, and they can eventually reach consensus no matter when exactly each node gets the conclusion. Gossip about gossip can make sure most of the asynchronized nodes can catch up to the latest hashgraph because all the events and gossip history can be available as soon as they restore synchrony. Asynchronized nodes won't miss anything just because they are unreachable for a period and can get back to the network and participate in voting as usual.

C. Finality

A concept called confirmation confidence is introduced in Tangle to indicate a transaction's level of acceptance by the rest of the Tangle. The calculation of confirmation confidence is rather simple:

1) Run tip selection algorithm 100 times;
2) Count how many of those 100 tip selections end up approving some specific transaction;
3) This transaction's confirmation confidence is the percentage of the previous step's result.

Confirmation confidence is a measure of how confirmed and unchangeable a transaction is. Clearly, if a transaction is on the path that most of the incoming transactions go through when selecting tips to approve, then the possibility to overthrow this transaction is significantly low because the attacker has to overthrow most of the transactions that approve it as well. Even though a transaction has a very high confirmation confidence, it means that this transaction is very hard to be pushed out of the consensus but not impossible. There is no rule in consensus to indicate the minimum confirmation confidence a transaction needs to reach to be considered as a part of irreversible history. In order to eliminate the little but possible chances that a highly confirmed transaction is later pushed out of the consensus, IOTA Foundation introduces a trusted centralized third-party: the coordinator. A milestone transaction is issued every two minutes by the coordinator and all the transactions approved by the milestone transaction are considered to have a confirmation confidence of 100% and to be unchangeable, which is called "coordicide". IOTA Foundation says the coordinator is a temporary solution at early period and will be shut down when the IOTA is at an enough scale and fully decentralized. The IOTA Foundation claims to get rid of it eventually but currently no timeline is announced.

Finality in Hashgraph is relatively simple. Once the consensus is reached, finality is completely guaranteed, which means that there is nothing the attacker can do to change the confirmed event. A concept called round is defined in Hashgraph to divide time into different segments. Roughly speaking, it will be in a new round if one of the nodes is confident enough that the supermajority of the nodes in the network has the same vote to an election. Order of events confirmed in a round cannot be changed in the following rounds and the transactions in previous rounds can be considered as irreversible ones.

D. Performance

IOTA does not have a bound for transaction rate thanks to its chainless design to get free of ordering transactions and mining. In theory, Tangle's performance will only be limited by physical hardware, namely bandwidth to propagate transactions. Normally, IOTA has a transaction rate under 20 tps according to live transaction monitor provided by thetangle.org. A peak transaction rate more than 400 tps has been seen in IOTA before and it has been handled well with a confirmation ratio over 99%. Since most of the users will follow the encouraged tip selection algorithm, it will not take too long for a new transaction to be approved as long as the bandwidth and computing power is sufficient to propagate transactions and make approvals. Even if there are peak hours when the transaction rate is relatively high and several different transactions are more likely to approve the same tip and consequently leave some other tips slightly

behind, it is allowed to promote the left transactions with an additional empty transaction. This makes the long delays that happen with the accidentally left transactions quite rare. As for the time it takes for an approved transaction to be irreversible, it depends on the rate of incoming transactions, the higher it is, the faster the approved transactions will reach the finality. The performance of IOTA is currently sufficient for most of the public blockchain networks.

Modern BFT protocols have been proved to sustain tens of thousands of transactions in both prototypes and practical systems [18]. Hashgraph is a BFT protocol that does not even need the message exchange among nodes to reach final consensus. According to its whitepaper [11], Hashgraph's throughput is only limited by bandwidth. Even a fast domestic Internet connection in Hashgraph is sufficient to handle all the transactions of an entire worldwide VISA network. Simulations in the whitepaper show that there are clear tradeoffs between throughput, latency, number of nodes and geographic distribution. The performance varies depending on processing power and transaction size. Normally, throughput ranges from 50,000 tps up to 500,000 tps with a latency varying from 0.04 seconds to 11 seconds in a network with less than 1000 users. The nature of Hashgraph makes it used more for a consortium blockchain network than a public one.

E. Tolerated Power of an Adversary

To initiate an attack in IOTA, the attacker, roughly speaking, should be capable of issuing, propagating and approving transactions faster than the rest of the network combined. To launch a double-spending attack, for instance, the attacker firstly sends a normal transaction. It will later be approved by the milestone transaction and the attacker then issues a conflicting transaction that apparently will not be approved by honest users. To make it approved and confirmed, the attacker starts issuing as many transactions as he can to approve the conflicting transaction. If the attacker has enough computation power to send more transactions than everyone else combined, the cumulative weight of conflicting transaction's branch will increase faster than any other branches and consequently be followed by the new incoming transactions. At some level, the conflicting transaction will be the valid one because it has a higher confirmation confidence. Currently, initiating a double-spending attack is basically impossible due to the coordinator who issues a milestone transaction every two minutes to make all the transactions approved by it irreversible with a confirmation confidence of 100%. It is expected that the network will be strong enough to resist attacks like this in the future and the coordinator will be taken down.

As depicted in Fig. 3, Bob who intends to initiate a double-spending attack in Hashgraph illegally forks his own ancestor history to create two conflicting events with the same self-ancestor and subsequently gossips to Alice and Carl. The honest users, Alice and Carl in Fig. 3, will eventually notice the trick through gossip about gossip and they will ignore the conflicting events with the same self-ancestor. If the attacker gossips an event that conflicts with the Hashgraph history, it will be considered as invalid and will be abandoned. Generally, gossip about gossip can make the communication history visible to all the nodes in the network, initiating an attack is basically impossible as long as the nodes can gossip with each other. Hashgraph has been proven to be an algorithm with Byzantine fault tolerance and the tolerated number of attackers is less than one-third of the total amount. However, Hashgraph allows being modified to avoid Sybil attack by making users unequal in virtual voting. A consensus mechanism similar to proof-of-stake (PoS) can be applied, where each user is assumed to have some positive integer associated with them, known as their stake. When voting, user's vote is weighted proportionally to the voter's stake. In this case, the tolerated number of attacks will be a set of users whose total stake is less than one third.

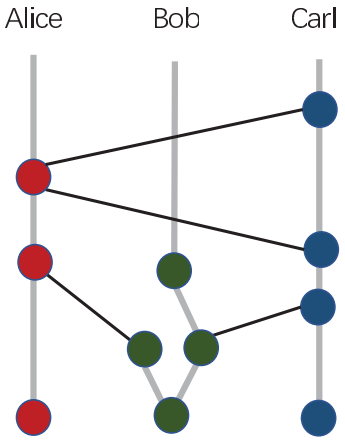

Fig. 3. An example of double-spending attack in Hashgraph

F. Scalability

Great scalability is always an outstanding merit of DAG-based cryptocurrencies. IOTA allows and encourages more users to join the system as it makes the system more stable by raising the difficulty to initiate attacks. Unlike blockchain which requires all the transactions to be ordered and join a queue to be confirmed, transactions in Tangle can be confirmed simultaneously. Chainless design in Tangle can make the number of tips grow enormously and subsequently make the approval of these tips very time-consuming because too many tips need a huge number of incoming transactions to approve all of them, which does not happen all the time. When there is fluctuation in the amount of the incoming transactions, the time it takes to digest the tips generated at the peak moment could be quite long if there is no bound for the number of tips. IOTA, in its whitepaper, assumes the total number of tips in the system remains stable and positive recurrent [19]. It also expects the total number of tips should fluctuate around a constant value and is not supposed to escape to infinity, otherwise many unapproved transactions would be

left behind. Tangle provides several tip selection algorithms to tackle the "overwhelming tips issue" that might be caused by the increasing number of users.

Hashgraph does not need bandwidth for message exchange and consequently makes it more unencumbered than IOTA when the network's load is at a relatively high level. Great capacity of throughput is one of the most distinguished features in Hashgraph. It is shown in simulations provided in whitepaper that the growing number of nodes in Hashgraph does not have a huge impact on throughput. It might increase the delay but the latency is still in seconds, which is tolerable in most of the cases. The number of Hashgraph users is tolerable within the size of a consortium blockchain network, its scalability in a public network with over millions of users or even higher is still inferior compared with IOTA.

G. Incentive

One of the merits of Tangle is that it is free of transaction fee or mining. Considering most of the users might issue their own transactions, it is particularly advantageous for them to propagate new transactions that approve their own transactions. Even though there might be users who have never issued any transaction recently, Tangle still manages to inspire them to actively participate in transaction propagation. Each user calculates how many transactions are received from a neighbor. If one particular node is "too lazy", it will be dropped by its neighbors and completely blind in the Tangle. Therefore, even for a user who has never issued any transaction, it still has an incentive to participate or it can be challenging to keep up with the network.

Hashgraph is an algorithm that emphasizes little on incentive. However, Hedera Hashgraph, a cryptocurrency and public distributed ledger based on the Hashgraph algorithm, introduces a more detailed scheme for operation including incentives for nodes to join and participate. Generally speaking, operating nodes can receive fees from maintaining the system. An algorithm similar to PoS is applied to weight the votes of a particular node based on the amount of the currency it owns. Users in Hedera Hashgraph have to pay fees to transfer cryptocurrency, so it will compensate for the computing, bandwidth and storage resources operating nodes use in establishing consensus and providing services. Overall, there are various options for incentives in Hashgraph depending on the scenario it is applied in. As for the Hedra Hashgraph, weighted vote and earning fee are the predominant incentives for users to participate and maintain the system.

VI. Conclusion

In this paper, we have summarized the key features of two DAG-based models, IOTA and Hashgraph. DAG-based models are expected to provide great scalability and fast economical payments that most of the blockchain-based platforms fail to achieve. Hence, DAG is considered to be an ideal solution for the exponential growth in scale of IoT ecosystems. A set of properties are analyzed for the evaluation of two representative DAG-based models in IoT scenarios to explore the feasibility of their consensus mechanisms to handle the issues current IoT ecosystems are facing. Further studies need to be conducted to explore a possible modular DAG-based design for various IoT ecosystems with great acceptability.

References

[1] Felix Wortmann and Kristina Flüchter. Internet of things. *Business & Information Systems Engineering*, 57(3):221–224, 2015.

[2] Sabrina Sicari, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini. Security, privacy and trust in internet of things: The road ahead. *Computer networks*, 76:146–164, 2015.

[3] Shameek Bhattacharjee, Mehrdad Salimitari, Mainak Chatterjee, Kevin Kwiat, and Charles Kamhoua. Preserving data integrity in iot networks under opportunistic data manipulation. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pages 446–453. IEEE, 2017.

[4] Rodrigo Roman, Jianying Zhou, and Javier Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266–2279, 2013.

[5] Dennis Miller. Blockchain and the internet of things in the industrial sector. *IT Professional*, 20(3):15–18, 2018.

[6] Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. 2008.

[7] Sorin Zoican, Roxana Zoican, Marius Vochin, and Dan Galatchi. Blockchain and consensus algorithms in internet of things. In *2018 International Symposium on Electronics and Telecommunications (ISETC)*, pages 1–4. IEEE, 2018.

[8] Ali Dorri, Salil S Kanhere, and Raja Jurdak. Blockchain in internet of things: challenges and solutions. *arXiv preprint arXiv:1608.05187*, 2016.

[9] Sujit Biswas, Kashif Sharif, Fan Li, Boubakr Nour, and Yu Wang. A scalable blockchain framework for secure transactions in iot. *IEEE Internet of Things Journal*, 2018.

[10] Serguei Popov. The tangle. *cit. on*, page 131, 2016.

[11] Leemon Baird. The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance. *Swirlds Tech Reports SWIRLDS-TR-2016-01, Tech. Rep.*, 2016.

[12] Yoad Lewenberg, Yonatan Sompolinsky, and Aviv Zohar. Inclusive block chain protocols. In *International Conference on Financial Cryptography and Data Security*, pages 528–547. Springer, 2015.

[13] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[14] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR COMPUTER SCIENCE, 1982.

[15] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

[16] Florence Benezit, Patrick Denantes, Alexandros G Dimakis, Patrick Thiran, and Martin Vetterli. Reaching consensus about gossip: convergence times and costs. *Information Theory and Applications*, 2008.

[17] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. On bitcoin and red balloons. In *Proceedings of the 13th ACM conference on electronic commerce*, pages 56–73. ACM, 2012.

[18] Marko Vukolić. The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *International workshop on open problems in network security*, pages 112–125. Springer, 2015.

[19] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014.