B-IoT: Blockchain Driven Internet of Things with Credit-Based Consensus Mechanism

Junqin Huang*, Linghe Kong*, Guihai Chen*, Long Cheng†, Kaishun Wu‡ and Xue Liu§
*Shanghai Jiao Tong University, China, Email: {junqin.huang, linghe.kong}@sjtu.edu.cn, gchen@cs.sjtu.edu.cn
†Clemson University, USA, Email: lcheng2@clemson.edu
‡Shenzhen University, China, Email: wu@szu.edu.cn
§McGill University, Canada, Email: xueliu@cs.mcgill.ca

***Abstract*—Internet of Things (IoT) plays an indispensable role in our daily life, in many cases, IoT systems are implemented following the client-server paradigm, which are vulnerable to single point of failures and malicious attacks. Due to the resilience and security promise of blockchain, the idea of combining blockchain and IoT has gained considerable attention in recent years. However, blockchains are power-intensive and low-throughput, which may not suitable for power-constrained IoT devices. To tackle these challenges, we present B-IoT, a blockchain based IoT system with credit-based consensus mechanism. We propose a credit-based proof-of-work (PoW) mechanism for IoT devices, which enhances security and improves transaction efficiency simultaneously. In order to protect the confidentiality of sensitive IoT data, we design a data authority management method to regulate the access to sensor data. In addition, our system is built based on a directed acyclic graph (DAG)-structured blockchain, which is more efficient than the satoshi-style blockchain. We implement a prototype of B-IoT on Raspberry Pi, and conduct case studies of a smart factory. Extensive evaluation and analysis results demonstrate that the proposed credit-based PoW mechanism and data access control are practical for IoT.**

***Index Terms*—Internet of Things, blockchain, credit-based, proof-of-work, directed acyclic graph, security.**

I. Introduction

Internet of Things (IoT) is considered as a bridge to connect diverse devices together. By enabling easy access and interaction with a wide variety of devices, IoT has become an integral part of the Internet [1]. It is reported that the IoT smart objects are expected to reach 212 billion entities deployed globally by the end of 2020 [2], which offers a great market opportunity for equipment manufacturers, Internet service providers and application developers. In the present, many IoT systems are being applied in the healthcare service [3], transportation and logistics [4], firefighting [5], etc.

Even though the IoT technique has been applied in many fields, security attacks and failures can cause great trouble in the global IoT network [6], which may outweigh any of its benefits. For example, the central data center in the IoT system is vulnerable to the single point of failure [7] and malicious attacks such as DDoS, Sybil attack [8], which disrupt services availability. Besides, sensor data stored in central data center are at the risk of disclosure. In the meantime, data interception may occur in communications between IoT devices, which cannot promise the credibilities of collected data.

In recent years, with the emergence of blockchain, the idea of combining blockchain and IoT has gained considerable attention [9]–[12]. By leveraging the features of tamper-proof and decentralized consensus mechanism in blockchain, we have the chance to solve aforementioned security issues in IoT systems.

There are some existing research on this topic, for example, O. Novo [10] proposed an access control system based on blockchain technology to manage IoT devices. However, the system is not fully built on a distributed architecture because of the usage of the central management hub. Once the management hub is failed or attacked, IoT devices connected to it will be unavailable. Z. Li et al. [13] exploited the consortium blockchain technology to propose a secure energy trading system. But they did not consider privacy issue such as the sensitive data disclosure risk. Besides, they all adopt chain-structured blockchains in IoT systems, which overload power-constrained IoT devices. In the meantime, some other challenges arise when introducing the novel design of blockchain into IoT systems. We summarize three main challenges:

1) The trade-off between efficiency and security: We know that consensus algorithms in blockchain can effectively help to defend malicious attacks. PoW is the most widely used consensus algorithm, which forces nodes to run high complexity hash algorithms to verify transactions. However, it is overloaded for power-constrained IoT devices. While eliminating the PoW mechanism can potentially improve efficiency of transactions, it causes security issues. How to balance the trade-off between security and efficiency in consensus mechanism is the first challenge.

2) The coexistence of transparency and privacy: Blockchain features of transparency, which is an important characteristic in the finance field. However, it may become a drawback for some IoT systems, where the collected sensitive data require the confidentiality and are only accessible by authorized ones. Thus how to design the access control scheme in a transparent system is the second challenge.

3) The conflicts between high concurrency and low throughput: There are various IoT devices reporting data all the time in IoT systems, which demand high concurrency. Unfortunately, complex cryptographic based security mechanisms largely limit the throughput of blockchain. Besides, the synchronous consensus model in chain-structured blockchains

2575-8411/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDCS.2019.00135

cannot make full use of bandwidth in IoT systems. So how to improve the throughput of blockchain to satisfy the need of frequent transactions in IoT systems is the third challenge.

These three challenges motivate us to design a general, scalable and secure blockchain-based IoT system. To address these challenges, we propose a blockchain system with credit-based consensus mechanism for IoT, named B-IoT. In order to decrease the power-consumption in consensus mechanism, we present a self-adaptive PoW algorithm for power-constrained IoT devices. It adjusts the difficulty of PoW based on nodes' behaviours, which decreases the difficulty for honest nodes while increasing it for malicious nodes. We also present an access control scheme based on the symmetric cryptography in the transparent blockchain system, which provides a flexible data authority management method for users. Our system infrastructure is built based on the DAG-structured blockchain, which improves the system throughput by leveraging its asynchronous consensus model.

We implement a proof of concept system on Raspberry Pi for a case study of the smart factory. Extensive experiments and analysis results demonstrate that the proposed credit-based PoW mechanism and data authority management method can guarantee efficiency and security simultaneously. Our main contributions of this paper are described as following:

- We propose a general, scalable and secure blockchain system for IoT, where we design a moderate-cost credit-based PoW mechanism and an efficient access control scheme for power-constrained IoT devices.
- Different from previous works, we utilize the DAG-structured blockchain as the infrastructure instead of the chain-structured blockchain to build our system, which can achieve a high throughput.
- We design and implement a proof of concept system for a case study of the smart factory. The results of experiments show that the proposed credit-based PoW mechanism and the data authority management method are practical in IoT devices.

The remainder of this paper is organized as follows. Section II briefly introduces the background of blockchain technology. We describe our threat models in Section III. Section IV presents the overview of our blockchain system for smart factory including architecture and mechanisms design. In Section V, we implement a prototype of B-IoT and introduce the workflow of B-IoT in detail. Evaluation and analysis are conducted in Section VI. Section VII discusses the related work, and Section VIII concludes this paper.

II. Background

Blockchains are distributed ledgers or databases that enable parties which do not fully trust each other to form and maintain consensus about the existence, status and evolution of a set of shared facts, which is backed by complex cryptographic technologies and consensus models [14]. These values of blockchain have gained considerable interest and adoption in industry and academia.

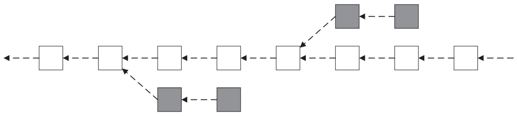


Fig. 1. Chain-structured blockchain. White squares represent valid blocks, and gray squares represent invalid blocks.

Blockchains can be sorted into two types by structures, one is chain-structured blockchain and the other is DAG-structured blockchain [15].

A. Chain-Structured Blockchain

Existing implementations of blockchain are mainly based on chain-structured blockchain, such as Bitcoin, Ethereum, Hyperledger. As Fig. 1 shows that chain-structured blockchain maintains the longest chain as the main chain in the system, blocks attached in the main chain are considered as valid transactions. When two blocks are generated just a few seconds apart, forks will happen, and the latest block in the longest chain is always chosen, so other blocks in shorter chains are considered as invalid blocks.

However, chain-structured blockchain is power-intensive due to its complex cryptographic security mechanisms [16], which is not suitable for power-constrained IoT devices. Also, synchronous consensus mechanisms limit the system throughput, i.e., transactions only can be validated one by one, which cannot satisfy the need for frequent requests in IoT systems.

B. DAG-Structured Blockchain

In order to make blockchain technology more practical in realistic world, especially in power-constrained applications, people propose a new structure of blockchain, based on directed acyclic graph architecture, known as *tangle* [17].

In tangle, it eliminates the concept of block, each transaction is an individual node linked in the distributed ledger. Before a new transaction is submitted, it must validate two former transactions that have been attached but not verified in the tangle, which is called *tips*. Then the new transaction bundles with these two former transactions through running PoW algorithm. After that, the new transaction can be broadcast to the tangle network. Each new transaction always will be validated by other newer transactions in later. There is a metric called *weight* for each transaction, which is proportional to the number of validation for the transaction. The weight is similar to the concept of six-block-security [18] in bitcoin, the larger value of weight is, the more difficult of the transaction to be tampered.

In chain-structured blockchain, a new transaction must be validated before attached to the main chain, which is called synchronous consensus. Different from it, tangle adopts an asynchronous consensus, which is more efficient in improving system throughput. As shown in Fig. 2, DAG-structured blockchain is not constrained by the single main chain and forks all the time. The relation among transactions looks like a tangled net. This novel architecture can improve network

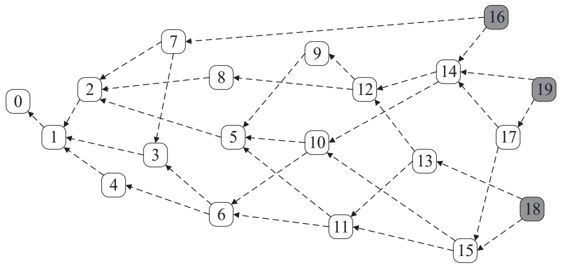


Fig. 2. Directed acyclic graph (DAG)-structured blockchain. White squares represent verified transactions, while gray squares represent tips.

throughput and system response time. IOTA, Byteball, NANO [19] are three representative DAG-structured blockchains.

C. Why Blockchain in IoT

The basic feature of blockchain is the decentralized trust, i.e., the non-tamperable source of data. Today, most IoT systems rely on the centralized data center to store massive data. Due to the inability to verify that data was not manipulated, IoT data cannot be fully trusted outside a owner's domain. For example, autonomous car startups and ride sharing giants, such as Uber and Lyft, have no solution to share trusted mapping or ride data. Instead, they gather and store similar datasets independently. How can we break down these monolithic data siloes and enable trust across parties? The combination of secure IoT devices and blockchain introduces a uniquely pure source of IoT data, which is known to come from a specific source and be non-manipulated [20].

In this paper, we adopt DAG-structured blockchains as the underlying technology. Though DAG-structured blockchains are designed for IoT or IoT-like systems, which contain computation capability limited devices, they still cannot satisfy the demands of frequent transactions because of its high complexity consensus algorithm. Hence, we need to design new light-weight consensus mechanisms for IoT systems.

III. Threat Model

In this work, we assume that attackers are able to launch the following attacks. We are not concerned about how attackers launch different attacks, but focus on defending the system against these possible attacks.

- *Single Point of Failure*. The single point of failure is a part of a system whose failure will stop the entire system from working, which is undesirable in any system with a goal of high availability or reliability.
- *Sybil Attack*. In a peer-to-peer network, each node has one identity generally. There may exist evil nodes, which pretend multiple identities illegitimately, attempts to control most nodes in the network to eliminate the function of redundant replicated nodes, or to defraud multiple rewards, which is known as Sybil attack.
- *Lazy tips*. A 'lazy' node could always verify a fixed pair of very old transactions, while not contributing to the verification of more recent transactions. For example, a malicious entity can artificially inflate the number of tips by issuing many transactions that verify a fixed pair of transactions. This would make it possible for future transactions to select these tips with very high probability, abandoning the tips belonging to honest nodes effectively.
- *Double-spending*. A malicious node wants to spend the same token twice or more through submitting multiple transactions before the previous one is verified. Even though such behaviour will be detected and canceled by asynchronous consensus mechanism, it slows down the efficiency of system because other associated transactions also will be redone.

The role of the manager in our system is assumed trustful, our security analysis are conducted under this premise. In addition, we assume that the attackers have limited computation capability, whose computation capability is close to IoT devices in the system. We assume attacks from the hardware layer are out of our scope in this work. So the private keys stored in IoT devices are assumed secure.

Under the above threat model, B-IoT has three security and privacy goals: service availability, anti-attack capability and data confidentiality.

Service Availability. B-IoT can handle the single point failure and keep the system service available.

Anti-attack Capability. B-IoT can mitigate the effects of Sybil attack, lazy tips, double-spending, DDoS attack, etc.

Data Confidentiality. Sensitive sensor data in B-IoT will not be leaked, though these data are stored in the public blockchains.

IV. B-IoT: Blockchain Driven Internet of Things with Credit-Based Consensus Mechanism

In this section, we present the overview of the proposed blockchain-based IoT system. We will introduce the detailed system design from three parts, which are system architecture, credit-based PoW mechanism and data authority management method. We present system architecture designed for the case study of smart factory firstly.

A. Architecture Design for Smart Factory

The system is built on a DAG-structured blockchain, each entity is a node in the blockchain-based IoT system. In terms of functional division, they can be divided into two categories, light nodes and full nodes: Light nodes are those power-constrained devices like IoT devices. They do not store blockchain information due to their constrained nature. What they can do are to verify tips, run PoW consensus algorithm and send new transactions to full nodes. Full nodes are those more powerful devices like gateways or servers, their main duty is to maintain the whole blockchain network, i.e., the tangle. They receive transaction requests from light nodes and broadcast in the blockchain network to complete the transactions.

The architecture of our system is shown in Fig. 3, and there are four components.

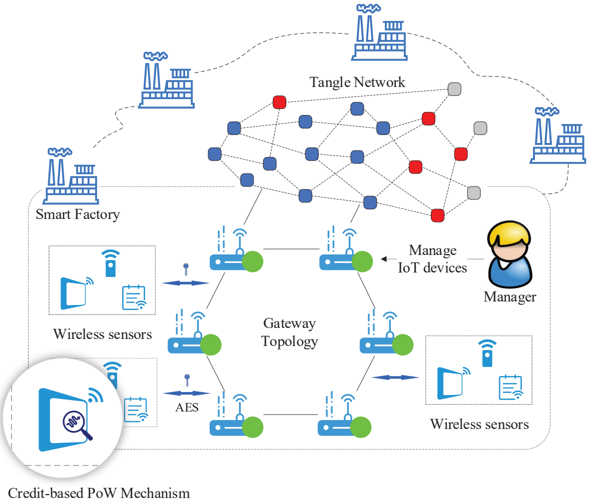


Fig. 3. The architecture of blockchain-based IoT system for smart factory.

1) Wireless sensors: Wireless sensors deployed in a smart factory belong to the group of light nodes. Each sensor will generate a blockchain account when initialized, i.e., a pair of public/secret key (PK, SK), which is the unique identifier in the system. The key pair for each device is not only used to sign transactions, but also to make the key distribution, which will be described in Section III-C.

2) Gateways: Gateways play the role of full nodes, which are committed to maintaining the tangle network. More specific, gateways receive the requests from various sensors, verify and broadcast the transactions in the tangle, they only process transactions from legal sensors that are authorized by the manager.

3) Manager: Manager is a specific full node, which is responsible for managing IoT devices in a smart factory. The public key of the manager will be hard-coded into genesis config of blockchain, which means only the manager has the rights to publish or update the authorization list of devices. Then the manager can manage IoT devices (authorize/deauthorize) through posting a new transaction where records public keys of authorized IoT devices. It can be described as:

$$TX = Sign_{SK_M}(PK_{d_1}, PK_{d_2}, ..., PK_{d_n}), \qquad (1)$$

where TX represents a transaction, SK_M represents the secret key of the manager, $PK_{d_1}, PK_{d_2}, ..., PK_{d_n}$ represent public keys of IoT devices. Because the manager signs the transaction by using his secret key, which cannot be forged, thus gateways can discriminate legal devices by fetching authorized devices list published by the manager from blockchain.

In each smart factory, the existence of one or more managers are permitted. The role of manager can help to manage the IoT devices in a smart factory. By leveraging the authorization list on blockchains, gateways can also block the requests from unauthorized devices. In this way, our system can be scaled and managed flexibly. In B-IoT, we consider that the manager is always a honest node.

4) Tangle network: The tangle network in our system is a public blockchain network, any party can access the network. Gateways, i.e., full nodes, keep the network secure and stable by broadcasting transactions and keeping copies of the blockchain.

Among factories, secure data sharing is also supported. For example, if factories need to configure their machines operating parameters for processing a certain kind of parts, they do not need to debug machines independently. They can request solutions of the same parts from other factories which have configured them through B-IoT. Due to the non-tamperable and traceable data on blockchains, B-IoT can break down these monolithic data siloes and enable trust across factories.

For some sensitive data, we can use data authority management method to protect the privacy of sensor data, which will be detailed introduced in Section III-C.

The architecture of our system is distributed and resilient to various attacks, such as DDoS, Sybil, double-spending, etc. Also, our system is based on DAG-structured blockchain, which improves system throughput comparing to chain-structured blockchain. In order to further improve throughput of our system and make access control in the system, we propose a credit-based PoW mechanism and a data authority management method in the rest part of this section.

B. Credit-Based PoW Mechanism

We design a credit-based PoW mechanism to balance the trade-off between efficiency and security in consensus mechanism.

We define that node i has a property of credit value Cr_i, and the credit value will change in real time based on node's behaviours. Normal behaviours, i.e., obey the system rules to send transactions, will increase the credit value over time gradually. In the opposite, nodes which conduct abnormal behaviours will decrease credit value. The difficulty of PoW mechanism is self-adaptive according to credit value of each node, the lower credit value is, the longer time taken to run PoW algorithm. So this mechanism will let honest nodes consume less resources while force malicious nodes to increase the cost of attacks.

Here we take two abnormal behaviours into consideration in designing of credit-based PoW mechanism, *lazy tips* and *double-spending*, which have been introduced in Section III.

Thus, according to the behaviour of node i, we divide Cr_i into two components, which can be denoted as:

$$Cr_i = \lambda_1 Cr_i^P + \lambda_2 Cr_i^N, \qquad (2)$$

where Cr_i^P represents the positive impact part, Cr_i^N represents the negative impact part, λ_1 and λ_2 represent the weight coefficient of each part respectively.

We can distribute the weight of these two parts by adjusting λ_1 and λ_2. If we want to adopt strict punishment strategy in the system, we can set λ_2 larger.

Cr_i^P is positively related to the number of normal transactions over a unit of time of node i, i.e., is measured by the level of node activity, which is defined as:

$$Cr_i^P = \frac{\sum_{k=1}^{n_i} w_k}{\Delta T}, \tag{3}$$

where n_i denotes the number of valid transactions of node i during the latest unit of time, ΔT denotes a unit of time, w_k denotes the weight of the k-th transaction. The weight of a transaction means the number of validation to this transaction.

That is to say, if node i is active during a period of time, Cr_i^P will adjust according to the level of activity, which guarantee active nodes in the system can submit transactions faster while using less power. If node i does not submit transactions for a period of time, we consider it as an inactive, even an untrusted node, so the system will not decrease the difficulty of PoW for it at the beginning, i.e., $Cr_i^P = 0$.

Cr_i^N is negatively related to the number of malicious behaviours of node i, which is defined as:

$$Cr_i^N = -\sum_{k=1}^{m_i} \alpha(\mathcal{B}) \cdot \frac{\Delta T}{t - t_k}, \tag{4}$$

where m_i represents the total number of malicious behaviours conducted by node i, t represents current time, t_k represents the time point of the k-th malicious behaviour conducted by node i, and $\alpha(\mathcal{B})$ represents the punishment coefficient for malicious behaviour $\mathcal{B}$, which is defined as:

$$\alpha(\mathcal{B}) = \begin{cases} \alpha_l & \textit{if } \mathcal{B} \textit{ is lazy tips behaviour;} \\ \alpha_d & \textit{if } \mathcal{B} \textit{ is double-spending behaviour,} \end{cases} \tag{5}$$

where α_l and α_d can be adjusted according to the requirement of sensitivity to malicious behaviours. We will discuss concrete parameters setting in Section V-A.

As described in Eqn. 4, we can observe that the impact of malicious behaviours on a node will gradually decrease over time. But different from Cr_i^P, the impact cannot be eliminated over time. When a malicious behaviour happened just a moment, the absolute value of Cr_i^N will be so large that the malicious node cannot continue conducting attacks because of the large difficulty of PoW. Thus we can mitigate the malicious behaviours in time. Because the credit value is calculated based on transaction weight and abnormal behaviours, which can be reflected from blockchain records, so the credit value cannot be forged or tampered.

After we calculate Cr_i^P and Cr_i^N respectively, we can get Cr_i according to Eqn. 2. And we define $Cr_i \propto \frac{1}{D_i}$, where D_i is the difficulty of PoW for node i.

The following part is to introduce how can we control the difficulty value of PoW algorithm. In tangle, a new transaction should bundle with two former transactions through PoW algorithm before submitting, which can be expressed as:

$$output = hash\{hash(TX_1)||hash(TX_2)||nonce\}, \tag{6}$$

where TX_1 and TX_2 are hash values of two former transactions respectively, the $nonce$ is a random number which nodes need to calculate. If $output$ satisfies the requirement of minimum length of prefix zero, then nodes succeed to find the valid nonce.

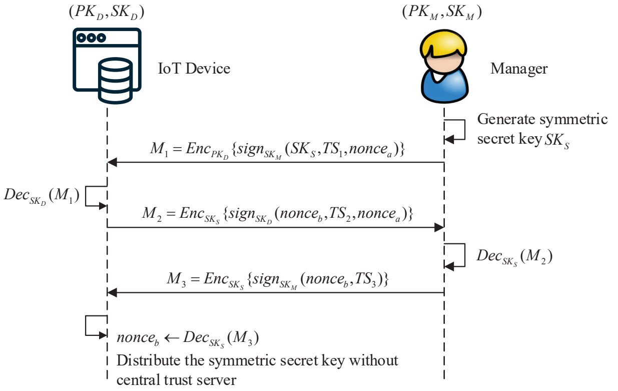

(PK_D, SK_D)
(PK_M, SK_M)
IoT Device
Manager
Generate symmetric secret key SK_S
$M_1 = Enc_{PK_D}\{sign_{SK_M}(SK_S, TS_1, nonce_a)\}$
$Dec_{SK_D}(M_1)$
$M_2 = Enc_{SK_S}\{sign_{SK_D}(nonce_b, TS_2, nonce_a)\}$
$Dec_{SK_S}(M_2)$
$M_3 = Enc_{SK_S}\{sign_{SK_M}(nonce_b, TS_3)\}$
$nonce_b \leftarrow Dec_{SK_S}(M_3)$
Distribute the symmetric secret key without central trust server

Fig. 4. The process of symmetric secret key distribution.

Due to the computation complexity and anti-collision of hash algorithm, we know that if the demand of minimum length of prefix zero is larger, it is more difficult to calculate a valid nonce. Thus we can control the difficulty of PoW through adjusting the demand of minimum length of prefix zero of the target hash string.

Hence, credit-based PoW mechanism can decrease the power consumption of honest nodes while mitigate malicious attacks efficiently.

C. Data Authority Management Method

Due to the transparency of blockchain, sensor data stored in blockchain is exposed in public. So we propose a data authority management method to support access control of sensor data in the system.

The way to protect data confidentiality in a transparent system is encryption. There are two main types of encryption algorithms, which are symmetric key encryption and public key encryption. Considering the efficiency of encryption algorithms, symmetric key encryption is much faster (about 100˜1000 times faster) than public key encryption, which is beneficial for power-constrained devices. Also, there are massive quantities of sensor data in smart factories, it is unbearable to use the much slower public key encryption.

However, different from public key encryption, if we adopt symmetric key encryption, we must consider a secure way to distribute the secret key. So in order to design a flexible data authority management method, we propose our secret key distribution scheme without any central trust party firstly.

From aforementioned architecture design, we know that every node has a pair of public/secret key (PK, SK) as the unique identifier, so we can utilize public key encryption to distribute the symmetric key.

There are three steps for one time secret key distribution, the process of secret key distribution is shown in Fig. 4, where TS denotes a timestamp, M denotes a message, Enc and Dec are the abbreviation of encrypt and decrypt respectively. The step of generating symmetric secret key is only done for one time. Each message is signed with the sender's secret key, which ensures the received message is not tampered or damaged. TS in each message presents timeliness of the message, which is used to resist replay attack.

M_1 is encrypted by the public key of IoT device, which means the message only can be decrypted by the IoT device. $nonce_a$ attached in M_1 is used to launch a response-challenge, if IoT device returns the correct nonce, we consider the IoT device has decrypted M_1 correctly. IoT device decrypts M_1 and gets the symmetric secret key, then sends M_2 encrypted by SK_S to demonstrate the success of decryption. $nonce_b$ is also a response-challenge which is used to test the correctness of SK_S. And manager returns $nonce_b$ in M_3 to complete this round of key distribution.

This key distribution scheme utilizes the public/secret key of each node to distribute symmetric secret key without any central trust server. Also, it is flexible to update symmetric keys if needed.

The function of each device is relatively fixed. For those devices whose collected non-sensitive data, they do not need to encrypt sensor data. So the manager only distributes secret key to those devices which collect sensitive data. After IoT devices get the symmetric secret key, then they can encrypt sensor data before posting it to blockchain. Only people who have the secret key can decrypt those sensitive data, which guarantees data confidentiality in a transparent system efficiently.

V. Implementation

In this section, we present the detailed implementation of our system. We use IOTA as the blockchain technology for the system, which is one of the most popular DAG-structured blockchain platform currently. In the rest part of this section, we will introduce the implementation of full nodes and light nodes.

A. Full Nodes

There are two roles of full nodes, manager and gateway. They are implemented based on IRI[1], which is the official reference implementation of full nodes. A full-featured node is a part of the tangle network as both a transaction relay and network information provider. It provides a convenient RESTful HTTP interface, so light nodes can post transactions to full nodes through the RPC interface. Besides, for the functionality of symmetric key generation and distribution, we use SHA-256 algorithm built in IOTA to distribute secret key, and use AES block cipher algorithm implemented by C to encrypt sensor data.

B. Light Nodes

Light nodes are IoT devices in this system, which connect to full nodes to interact with the tangle network. They are implemented based on PyOTA[2], which is the IOTA Python API Library. However, PyOTA does not provide local PoW interface, in order to adjust the difficulty of PoW algorithm flexibly, so we implement an extension package written in Java to extend PyOTA. The package is implemented according to the proposed credit-based PoW mechanism. We also implement AES-based data authority management method on light nodes by using C to encrypt collected sensor data.

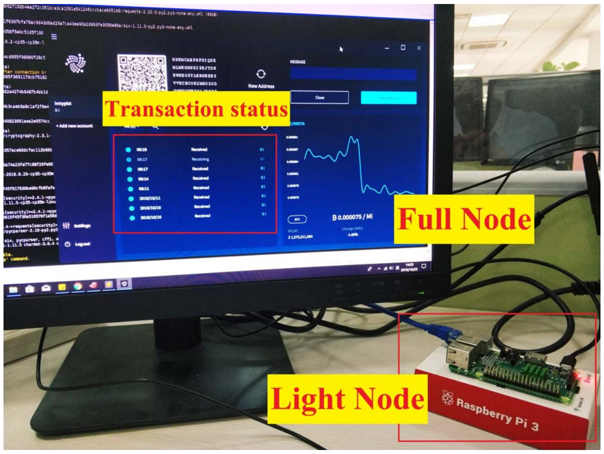


Fig. 5. The B-IoT prototype implemented on Raspberry Pi and PC.

C. Tangle Network

Full nodes maintain the tangle network through broadcasting, storing and synchronizing blockchain information, and light nodes contribute to increasing the stability of tangle through validating and submitting new transactions. Here we use a PC as a gateway/manager to run a full node, and use a Raspberry Pi Model 3B as an IoT device to run a light node, which is shown in Fig. 5. The Raspberry Pi reports collected data continuously and the PC screen shows the status of transactions in real time.

In this system, the interaction between manager, gateway and IoT device is shown in Fig. 6. The workflow of system can be described as following steps:

1) The manager initializes gateways to set up the tangle network firstly, i.e., records gateways identifiers in blockchain that cannot be tampered.
2) Then, the manager can authorize or deauthorize IoT devices through updating authorized devices list in the form of a transaction.
3) In the stage of secret key distribution, the manager does not need to distribute secret key to all IoT devices, only to devices which collect sensitive data. More specific, in this case, for IoT device 1, it does not need to encrypt collected sensor data because its data is not sensitive, but for IoT device 2, it will encrypt data by using symmetric secret key before posting transactions in order to guarantee sensitive data privacy.
4) After that, an IoT device will get two random tips to validate them before submitting a new transaction.

[1] [Online]. Available: https://github.com/iotaledger/iri
[2] [Online]. Available: https://pyota.readthedocs.io/

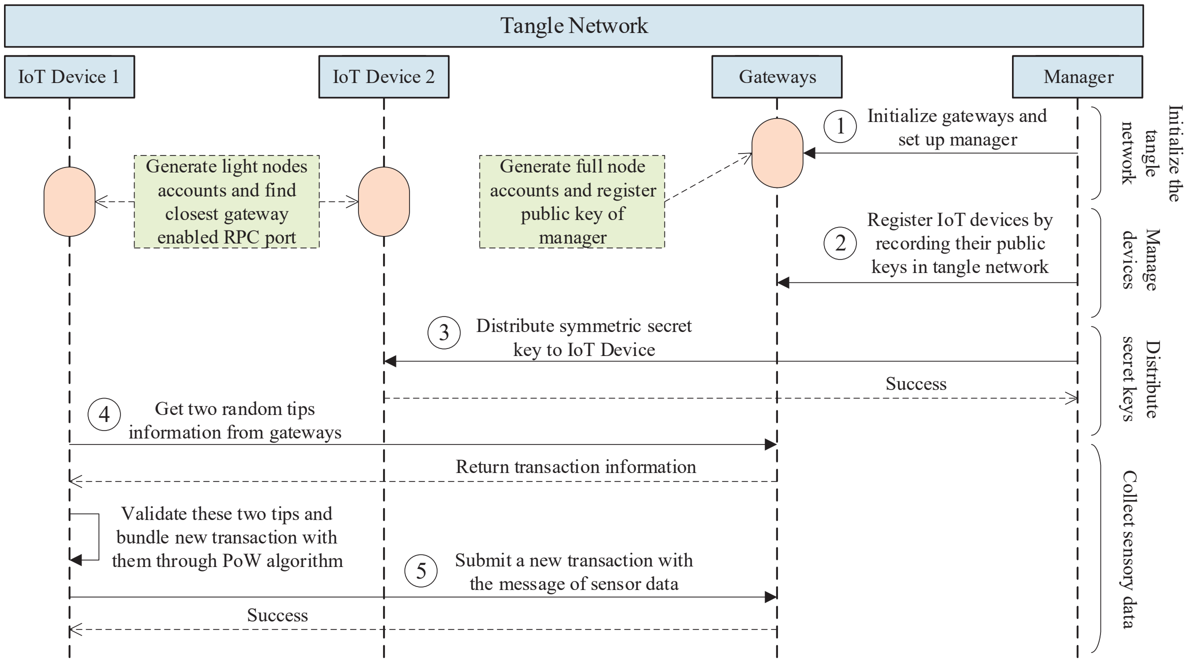

Tangle Network
IoT Device 1
IoT Device 2
Gateways
Manager
1 Initialize gateways and set up manager
Generate light nodes accounts and find closest gateway enabled RPC port
Generate full node accounts and register public key of manager
2 Register IoT devices by recording their public keys in tangle network
3 Distribute symmetric secret key to IoT Device
Success
4 Get two random tips information from gateways
Return transaction information
Validate these two tips and bundle new transaction with them through PoW algorithm
5 Submit a new transaction with the message of sensor data
Success
Initialize the tangle network
Manage devices
Distribute secret keys
Collect sensory data

Fig. 6. The interaction among manager, gateway and IoT device.

5) When validation is passed, the IoT device bundles the new transaction with these two verified tips through PoW algorithm, and submits it to the gateways.

Step 4 and step 5 are just an one-time process of sensor data submission, which can be done repeatedly.

VI. Evaluation and Analysis

In this section, we evaluate the performance in credit-based PoW mechanism and how the data authority management method impact on the efficiency of transactions. In addition, we provide security analysis of the whole system from two perspectives, i.e., system security and privacy security. IOTA already provides an official live transaction visualizer[3], which also displays average number of transaction per second (TPS) of the whole tangle network. For this reason, this section will not evaluate tangle network and target the new components proposed in our system.

Because the system is designed for IoT devices, in order to make evaluation results more convincing, all experiments were done on a Raspberry Pi Model 3B with Quad Core@1.2GHz, which is a power-constrained and computation-limited device.

A. Performance in Credit-Based PoW Mechanism

In this part, we evaluate the credit-based PoW mechanism comparing to the traditional PoW algorithm. We firstly discuss parameters setting that presented in Section III-B.

We carried out the PoW algorithm at different difficulty values, in order to find the relationship between the running time and the difficulty value of PoW. The result is shown in Fig. 7.

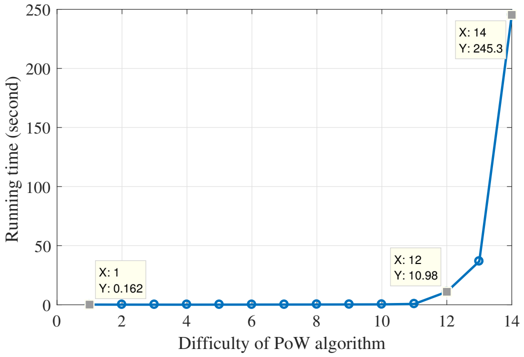
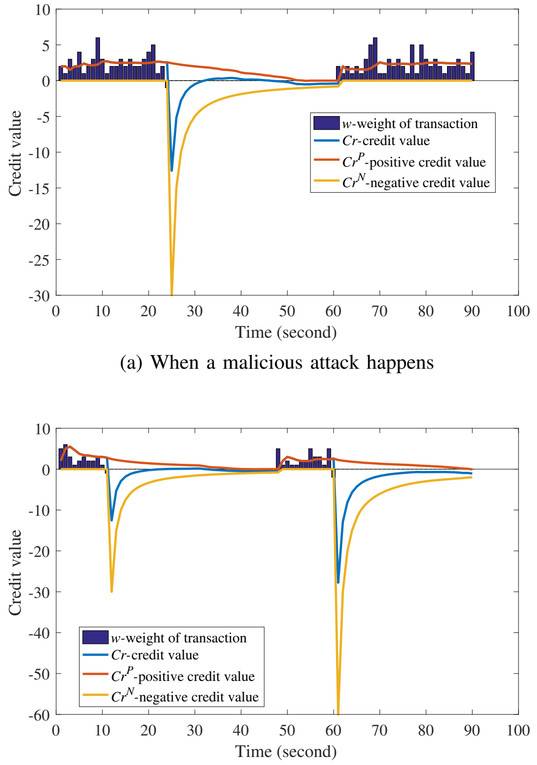


Fig. 7. Running time of PoW algorithm with increasing difficulty.

The minimum difficulty of PoW is 1, and the maximum should not exceed the length of hash. Indeed, it cannot reach the maximum value for normal light nodes because running time increases exponentially when the value of difficulty D is larger than 11, and when $D = 14$, the running time on Raspberry Pi has reached 245.3 seconds, which is unbearable. But on the other side, it is also a good way to punish malicious nodes.

Due to the running time of PoW on different IoT devices may be different, in this experiment, we choose the range of difficulty from 1 to 14. We set 11 as the initial difficulty of PoW, which is the relatively appropriate initial value for computation capability limited IoT devices, i.e., the difficulty is not too large also not too small.

[3] [Online]. Available: https://thetangle.org/live



(a) When a malicious attack happens

(b) When two malicious attacks happen

Fig. 8. Credit value changes based on nodes' behaviours.

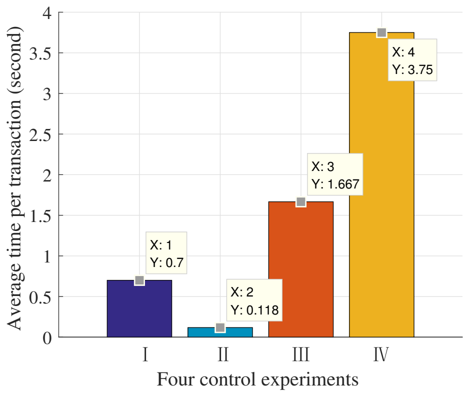


Fig. 9. Performance evaluation in credit-based PoW mechanism. The initial difficulty of PoW is 11. The four control experiments respectively represent *original PoW*, *credit-based PoW with normal behaviours*, *credit-based PoW with a malicious attack*, *credit-based PoW with two malicious attacks*.

In addition, according to Eqn. 2, there are four tunable parameters, which are λ_1, λ_2, ΔT and $\alpha(\mathcal{B})$. The weight of each transaction w can be counted from tangle network. Here we set $\lambda_1 = 1$, $\lambda_2 = 0.5$, $\Delta T = 30$ seconds, $\alpha(\mathcal{B}) = 0.5$ for the lazy tip event and $\alpha(\mathcal{B}) = 1$ for the double-spending event. We simulated behaviours of a light node to present working mechanism of credit-based PoW, which is shown in Fig. 8.

The x-axis represents sequence of time. We give a range of three ΔT to show how does credit-based PoW mechanism work. The y-axis represents credit value for three curves and denotes weight of transactions for bars. Also, we use a negative weight value to denote a malicious attack.

We can observe that the curve of Cr overlaps with that of Cr^P when $Cr^N = 0$, since the node does not conduct any malicious behaviour before, the negative credit part is 0. Once the node does any abnormal behaviour, and detected immediately. As a result, there will be a corresponding adjustment for credit value. From Fig. 8 (a), we can see that when time is at 24th second, the node conducts a malicious attack, Cr^N has a sharp decline in short time, thus Cr also sharply decreases according to Eqn. 2.

We know that $Cr \propto \frac{1}{D}$, which means the less Cr is, the more difficult PoW becomes. The node has to take a longer time to calculate a correct nonce for the next transaction after conducting a malicious behaviour. Thus there is a spacing between 24th second and 61st second in Fig. 8 (a) because of the punishment for the malicious behaviour. It takes 37 seconds to recover the normal transaction in this experiment, and during this time, Cr^P also decreases because it is inactive. The degree of punishment can be adjusted flexibly according to the requirement of system. As time goes, the credit value of node will increase gradually and return to normal transaction rate. Besides that, in Fig. 8 (b), if the node conducts malicious attacks twice or more, it will take longer time to recover normal transaction rate, which can well prevent malicious nodes from attacking. The simulation results indicate that the credit-based PoW mechanism can mitigate malicious attacks effectively.

Then, we compare the credit-based PoW mechanism with the original PoW mechanism, and set four control experiments as shown in Fig. 9.

We conducted these four control experiments during a range time of three ΔT, i.e., 90 seconds, and evaluate the average time of PoW per transaction. From Fig. 9, we can observe that the credit-based PoW with normal behaviours perform the best in running time, which only takes 0.118 second of PoW for each transaction on average, while it takes 0.7 second on average for original PoW mechanism. This indicates that the credit-based PoW can speed up transactions for honest nodes.

We also notice that for malicious nodes, the more malicious behaviours they conduct, the longer time they need to post a transaction. The penalty time is exponential with the number of malicious attacks, so malicious nodes can hardly complete a transaction which will consume much computing resources. The result indicates credit-based PoW mechanism can also defend malicious attacks efficiently even if an honest node becomes a malicious one suddenly.

B. Impact of Data Authority Management Method on Transaction Efficiency

We evaluated the data authority management method's influence on transaction efficiency. In the method, there mainly contains two components, which are secret key distribution

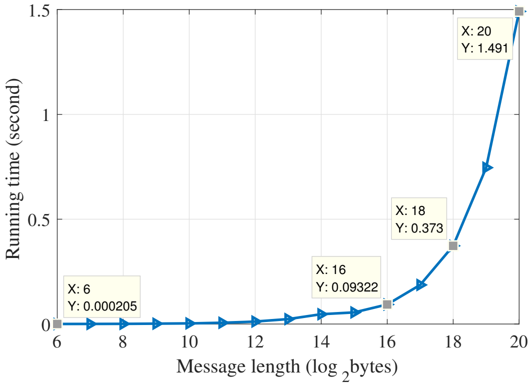


Fig. 10. Impact of symmetric encryption algorithm on transaction efficiency.

and sensor data encryption. Considering the frequency of use, key distribution will not be conducted frequently, even only conducted once at the initialization of system, impact on transaction can be ignored. Thus, in this part, we focus on evaluating performance in sensor data encryption.

As introduced in Section V, we adopt the AES algorithm in sensor data encryption. And we test the speed of data encryption for different message length, which is from 64 bytes to 1 millionbytes, and the result is shown in Fig. 10. Note that Fig. 10 uses a logarithmic scale.

We can observe that running time of AES increases with increasing message length. When message length is 64 bytes, the running time of AES is 0.205 millisecond. When message length is 1 millionbytes, the running time is 1.491 second. Indeed, a 256 kilobytes data package is large enough for IoT transmission. In this experiment, encrypting a message with 256 kilobytes length on Raspberry Pi only needs 0.373 second, which has tiny impact on the whole transaction process. Thus we can conclude that the data authority management method has reasonable impact on transaction efficiency.

C. Security Analysis

In this part, we analyze security from two aspects, which are system security and privacy protection respectively.

1) System security: We analyze system security from four threat models introduced in Section III, which are single point of failure, Sybil attack, lazy tips and double-spending. We discuss system security under the premise that the manager in B-IoT is always honest.

- **Single point of failure**. Because B-IoT is built based on DAG-structured blockchain, which is a distributed ledger, consisting of a group of replicated database nodes. Sensor data are redundantly replicated by all full nodes, so it is resilient for failure of one or more nodes, which improves reliability of IoT system.
- **Sybil attack**. In Section IV-A, we have already presented the blockchain-based devices management scheme. We know that information recorded in blockchain cannot be tampered, so we can leverage this feature to manage IoT devices by maintaining an authorization list on blockchain. And full nodes can decline to provide services for unauthorized IoT devices according to the list, which can effectively defend attacks like DDoS, Sybil attack.
- **Lazy tips**. Lazy tips behaviours can be detected easily according to verification records on blockchain, so the punishment mechanism in our proposed credit-based consensus algorithm can well defend from such attack by increasing the difficulty of consensus algorithm.
- **Double-spending**. The consensus mechanism in blockchain can prevent double-spending effectively, but the original consensus does not have punishment mechanism. Thus our proposed credit-based PoW mechanism in this work also helps to punish and defend from malicious nodes.

2) Privacy protection: In B-IoT, there are two groups of sensor data, sensitive and non-sensitive data. Due to the transparency of blockchain, it is necessary to protect data privacy for sensitive data.

As described in Section IV-C, we utilize symmetric encryption algorithm to implement a data authority management method, which provides sensor data confidentiality through encrypting data before storing in blockchain. Only users who have the secret key can decrypt and get sensor data, which regulates the access to sensitive sensor data in a transparent system. In addition, the data authority management method brings reasonable impact on transaction efficiency, which is resource-friendly to IoT devices.

VII. Related Work

In IoT system, there are common technical challenges [6], [21] needed to tackle such as scalability, dependability, privacy, access control, etc. In this section, we review related work for addressing these challenges and discuss the insufficiencies of them briefly.

There are existing solutions that are not based on blockchain technologies. For example, C. E. Kaed et al. [22] presented a semantic rules engine for IoT gateways that allows implementing dynamic and flexible rule-based control strategies, which is vulnerable to single point failure and malicious attacks due to the centralized architecture. M. Shamim Hossain et al. [23] presented a HealthIoT-enabled monitoring framework to collect healthcare data from mobile devices and sensors, which also faces the same risks. In addition, healthcare data stored in central servers may be vulnerable to privacy disclosure. Quaddah et al. [24] did a comprehensive survey of different access control solutions in IoT, they concluded that commonly used Internet protocols cannot be applied to constrained environments.

Thus, with the emergence of blockchain technology, researchers try to break traditional Internet protocols, i.e., client-server paradigm, and turn the research focus to applying blockchain to IoT to solve aforementioned issues. For example, A. Dorri et al. [9] proposed a Blockchain-based smart home framework to achieve security goals of confidentiality, integrity and availability. But they eliminated the concept of PoW to speed up efficiency of transactions, which will raise security risks. Z. Shae et al. [25] proposed a blockchain platform for clinical trial and precision medicine, which still stuck in the concept stage and is lack of evaluation. K. R. zylmaz et al. [26] tried to integrate low-power IoT devices to a blockchain-based infrastructure, but the system was implemented on Ethereum blockchain, which overloads IoT devices. And the low throughput of Ethereum blockchain cannot satisfy the demands of IoT system. Di Pietro et al. [27] described a distributed trust model for the IoT that bridges them to create end-to-end trust between IoT devices without any third party, which just applied the blockchain technology into IoT systems and did not present a detailed implementation.

VIII. Conclusion

In this work, we propose a blockchain-based IoT system to address aforementioned challenges for IoT. The proposed credit-based PoW mechanism, which decreases power consumption for honest nodes while increasing computing complexity for malicious nodes, helps to make the DAG structured blockchain more suitable for IoT systems. Also, the data authority management method can protect data privacy without affecting the system performance, which is also practical in IoT system. The results of extensive experiments and evaluation show that our system has a good performance in IoT.

This work will be of importance to research in distributed IoT systems by providing a practical DAG structured blockchain based solution. Our solution is not only suitable for the smart factory, but also able to applied in various IoT scenarios. However, there are still some limitations in our system, such as sensor data quality control, storage limitations. In future directions, we can explore sensor data quality control schemes in blockchain-based systems and some methods to store huge amounts of data.

Acknowledgment

This paper was partially supported by the National Key R&D Program of China 2018YFB1004703, NSFC grant 61672349 and 61672353.

References

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.

[3] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. Kwak, "The internet of things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[4] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[5] Z. Ji and Q. Anwen, "The application of internet of things (iot) in emergency management system in china," in *Technologies for Homeland Security (HST), 2010 IEEE International Conference on*. IEEE, 2010, pp. 139–142.

[6] Y. Lu and L. D. Xu, "Internet of things (iot) cybersecurity research: A review of current research topics," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[7] I. J. Vergara-Laurens, L. G. Jaimes, and M. A. Labrador, "Privacy-preserving mechanisms for crowdsensing: Survey and research challenges," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 855–869, 2017.

[8] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "Sybillimit: A near-optimal social network defense against sybil attacks," in *IEEE Symposium on Security and Privacy (S&P)*, May 2008, pp. 3–17.

[9] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2017, pp. 618–623.

[10] O. Novo, "Blockchain meets iot: An architecture for scalable access management in iot," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, April 2018.

[11] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[12] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[13] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3690–3700, Aug 2018.

[14] M. Swan, *Blockchain: Blueprint for a new economy.* " O'Reilly Media, Inc.", 2015.

[15] K. Karlsson, W. Jiang, S. Wicker, D. Adams, E. Ma, R. van Renesse, and H. Weatherspoon, "Vegvisir: A partition-tolerant blockchain for the internet-of-things," in *IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, July 2018, pp. 1150–1158.

[16] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Big Data (BigData Congress), 2017 IEEE International Congress on*. IEEE, 2017, pp. 557–564.

[17] S. Popov, "The tangle," *cit. on*, p. 131, 2016.

[18] R. Bhme, N. Christin, B. Edelman, and T. Moore, "Bitcoin: Economics, technology, and governance," *Journal of Economic Perspectives*, vol. 29, no. 2, pp. 213–38, May 2015.

[19] C. Bai, "State-of-the-art and future trends of blockchain based on dag structure," in *Structured Object-Oriented Formal Language and Method*. Springer International Publishing, 2019, pp. 183–196.

[20] IoTeX, "Blockchain & iot: What's it all about?" Oct 2018. [Online]. Available: https://hackernoon.com/blockchain-iot-whats-it-all-about-f594b3f0da1e

[21] K. Iwanicki, "A distributed systems perspective on industrial iot," in *IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, July 2018, pp. 1164–1170.

[22] C. E. Kaed, I. Khan, A. V. D. Berg, H. Hossayni, and C. Saint-Marcel, "Sre: Semantic rules engine for the industrial internet-of-things gateways," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 715–724, Feb 2018.

[23] M. S. Hossain and G. Muhammad, "Cloud-assisted industrial internet of things (iiot) enabled framework for health monitoring," *Computer Networks*, vol. 101, pp. 192–202, 2016.

[24] A. Ouaddah, H. Mousannif, A. A. Elkalam, and A. A. Ouahman, "Access control in the internet of things: Big challenges and new opportunities," *Computer Networks*, vol. 112, pp. 237–262, 2017.

[25] Z. Shae and J. J. P. Tsai, "On the design of a blockchain platform for clinical trial and precision medicine," in *IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, June 2017, pp. 1972–1980.

[26] K. R. zylmaz and A. Yurdakul, "Work-in-progress: integrating low-power iot devices to a blockchain-based infrastructure," in *International Conference on Embedded Software (EMSOFT)*, Oct 2017, pp. 1–2.

[27] R. Di Pietro, X. Salleras, M. Signorini, and E. Waisbard, "A blockchain-based trust system for the internet of things," in *Proceedings of the 23rd ACM on Symposium on Access Control Models and Technologies (SACMAT)*, 2018, pp. 77–83.