Towards Secure Industrial IoT: Blockchain System with Credit-Based Consensus Mechanism

Junqin Huang, Linghe Kong, *Senior Member, IEEE*, Guihai Chen, Min-You Wu,
Xue Liu, *Senior Member, IEEE*, Peng Zeng

***Abstract*—Industrial Internet of Things (IIoT) plays an indispensable role for Industry 4.0, people are committed to implementing a general, scalable and secure IIoT system to be adopted across various industries. However, existing IIoT systems are vulnerable to single point of failure and malicious attacks, which cannot provide stable services. Due to the resilience and security promise of blockchain, the idea of combining blockchain and IoT gains considerable interest. However, blockchains are power-intensive and low-throughput, which are not suitable for power-constrained IoT devices. To tackle these challenges, we present a blockchain system with credit-based consensus mechanism for IIoT. We propose a credit-based proof-of-work (PoW) mechanism for IoT devices, which can guarantee system security and transaction efficiency simultaneously. In order to protect sensitive data confidentiality, we design a data authority management method to regulate the access to sensor data. In addition, our system is built based on directed acyclic graph (DAG)-structured blockchains, which is more efficient than the satoshi-style blockchain in performance. We implement the system on Raspberry Pi, and conduct a case study for the smart factory. Extensive evaluation and analysis results demonstrate that credit-based PoW mechanism and data access control are secure and efficient in IIoT.**

***Index Terms*—Industrial IoT, blockchain, credit-based, proof-of-work, directed acyclic graph, security, efficiency, privacy.**

I. Introduction

THE integration of IoT and industry is an important modus to promote automation and informatization of industry. IIoT helps cut down on errors, reduce costs, improve efficiency and enhance safety in manufacturing and industrial processes, which has a great chance to make industry field a higher level of integrity, availability and scalability.

However, security attacks and failures could cause great trouble against the global IoT network [1], which may outweigh any of its benefits. For example, the central data center is vulnerable to single point failure and malicious attacks such as DDoS, Sybil attack [2], which cannot guarantee services availability. In addition, sensor data stored in a data center are at the risk of disclosure. Also, 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 interest [3]–[5]. By leveraging the features of tamper-proof and decentralized consensus mechanism in blockchain, we have the chance to solve the aforementioned security issues in IIoT systems.

There are some existing research on this topic, for example, O. Novo [4] proposes an access control system based on the 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 become unavailable. Z. Li et al. [6] exploit the consortium blockchain technology to propose a secure energy trading system. But they do not consider privacy issues such as the sensitive data disclosure risk, and thus it cannot guarantee sensitive data confidentiality. The aforementioned systems all adopt chain-structured blockchains in IoT systems, which are overloaded for power-constrained IoT devices. Z. Xiong et al. [7] introduce edge computing for mobile blockchain applications and present a Stackelberg game model for efficient edge resource management for mobile blockchain. They reduce computational requirements of mobile devices by leveraging edge computing. In addition, there are some other challenges that also brought in the meantime when introducing the novel design of blockchain into IIoT systems. We summarize three folds main challenges:

1) The trade-off between efficiency and security: We know that consensus algorithms in blockchain can effectively help to defend malicious attacks, and 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 system security issues. As a result, how to make the trade-off between security and efficiency in consensus mechanisms is the first challenge of this work.

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 IIoT systems, where the collected sensitive data require the confidentiality and are only accessible by authorized ones. It is therefore important to design an access control scheme in a transparent system.

3) The conflicts between high concurrency and low throughput: IoT devices report data continuously in IIoT systems, leading to a high concurrency. Unfortunately, complex

J. Huang, L. Kong, G. Chen, M.-Y. Wu are with Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: junqin.huang@sjtu.edu.cn; linghe.kong@sjtu.edu.cn; mwu@sjtu.edu.cn; gchen@cs.sjtu.edu.cn).

X. Liu is with McGill University, Montreal, QC H3A 0G4, Canada (e-mail: xueliu@cs.mcgill.ca).

P. Zeng is with the Laboratory of Networked Control Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China (e-mail: zp@sia.cn).

cryptographic based security mechanisms largely limit the throughput of blockchain. Besides, the synchronous consensus model in chain-structured blockchains cannot make full use of bandwidth in IIoT systems. So how to improve the throughput of blockchain to satisfy the need of frequent transactions in IIoT systems becomes the third challenge.

To address these challenges, we propose a blockchain system with credit-based consensus mechanism for IIoT. In order to decrease the power-consumption in consensus mechanism, we present a self-adaptive PoW algorithm for power-constrained IoT devices. It can adjust the difficulty of PoW based on nodes' behaviour, which can decrease the difficulty for honest nodes while increasing 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 concrete system on Raspberry Pi for a smart factory scenario. 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 identify three main challenges in integrating blockchain technology into IIoT and propose corresponding three solutions to tackle these challenges.
- We propose a general, scalable and secure blockchain system for IIoT, where we design a moderate-cost credit-based PoW mechanism and an efficient access control scheme for power-constrained IoT devices. Also, different from previous works, we utilize the DAG-structured blockchain as the infrastructure to build our system to achieve a higher throughput.
- We design and implement the proposed system for a smart factory scenario. Experiments results demonstrate that the credit-based PoW mechanism and data authority management method have a good performance in IoT devices.

The remainder of this paper is organized as follows. Section II briefly introduces the background of blockchain technology. Section III presents the overview of our blockchain system for smart factory including architecture and mechanisms design. We implement the proposed system in Section IV, and introduce the workflow of each system module respectively. Evaluation and analysis are conducted in Section V. Section VI discusses the related work, and Section VII concludes this paper.

II. Background

Blockchains are distributed ledgers or databases, which is backed by complex cryptographic technologies and the consensus model. Blockchains 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 [8]. These values of blockchain have gained considerable interest and adoption in industry and academia.

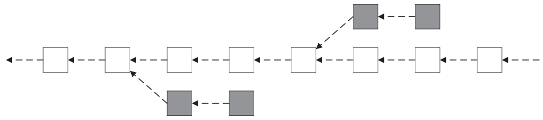


Fig. 1. Chain-structured blockchain. White squares represent valid blocks, while gray squares represent invalid blocks.

Based on the difference in structure, there are two types of blockchains, one is chain-structured blockchain and the other is DAG-structured blockchain [9].

A. Chain-Structured Blockchain

Existing implementations of blockchain are mainly based on chain-structured blockchain, such as Bitcoin, Ethereum, Hyperledger, etc. 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 [10], 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 application, people propose a new structure of blockchain, based on directed acyclic graph architecture, which is vividly called *tangle* [11].

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 [12] in bitcoin, the bigger value of weight is, the more difficult of a 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 and consensus mechanism

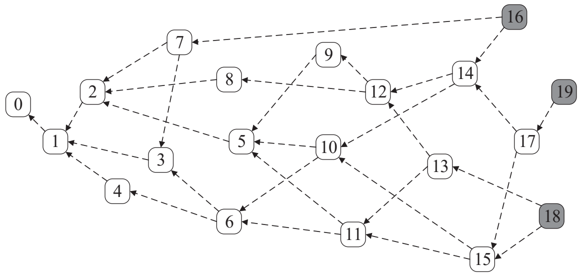


Fig. 2. Directed acyclic graph (DAG)-structured blockchain. White squares represent verified transactions, while gray squares represent tips.

can improve network throughput and system response time theoretically. IOTA [11], Byteball [13], NANO are three representative DAG-structured blockchains.

Though DAG-structured blockchain has been designed to satisfy the demands of frequent transactions in IoT system, ability-limited IoT devices, e.g., battery powered nodes, are restricted to run light wallets due to the complex consensus algorithm [14]. According to the official document, we know that the minimum difficulty value of proof-of-work required to attach transaction to tangle is 14[1], and we test its performance running on a Raspberry Pi in Section V. The Fig. 7 shows that it takes over 200 seconds to run the PoW algorithm, which is unacceptable for IIoT systems. Hence, we need to design a new light-weight consensus mechanism for IIoT systems.

III. A Blockchain System with Credit-Based Consensus Mechanism for IIoT

In this section, we present the overview of the proposed blockchain-based IIoT system. We introduce the detailed design of system from three parts, including the system architecture, credit-based PoW mechanism and data authority management method.

A. Architecture Design for Smart Factory

The system infrastructure is built on DAG-structured blockchain, each entity is a node in the blockchain-based IIoT system. In terms of functional division, they can be divided into two categories, i.e., 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 in the architecture.

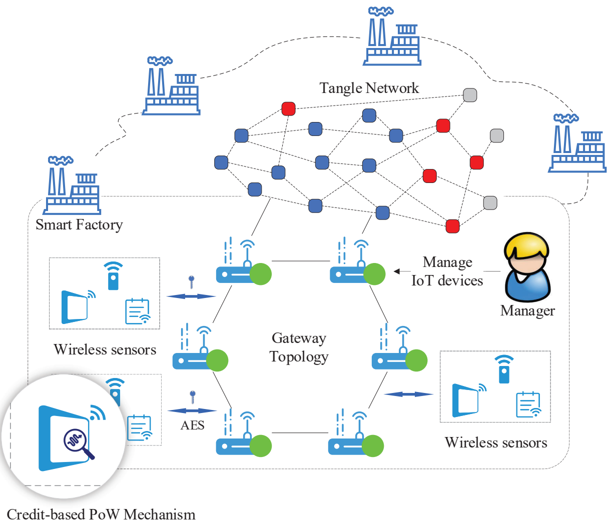


Fig. 3. The architecture of blockchain-based IIoT system for smart factory.

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 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 software in gateways, which means only the manager has the rights to publish the authorization list of devices. Then the manager can manage IoT devices (add/delete) through launching a 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, which depends on the decision of the owner of IoT devices. The role of a manager can help to manage the IoT devices in a smart factory, also block the invalid requests from unauthorized devices. In this way, our system can be scaled and managed flexibly.

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

[1] [Online]. Available: https://github.com/iotaledger/iota.js

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2903342, IEEE Transactions on Industrial Informatics

and stable by broadcasting transactions and keeping copies of the blockchain. Among factories, secure data sharing is also supported. 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 credit-based PoW mechanism and data authority management method in the rest part of this section.

B. Credit-Based PoW Mechanism

In this part, we design credit-based PoW mechanism to make 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.

Before giving the detailed design of credit-based PoW mechanism, we firstly state two possible existing abnormal behaviours in the system.

1) 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, effectively abandoning the tips belonging to honest nodes.

2) 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.

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

$$Cr_i = \lambda_1 Cr_i^P + \lambda_2 Cr_i^N, \tag{2}$$

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 bigger.

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 malicious behaviours impact on a node will gradually decrease over time, but different from Cr_i^P, it 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 stop the malicious behaviours in time.

We notice that the credit formulation mechanism requires full transaction information of each sensor involved, is it possible to calculate correct credit scores? We know that the whole tangle network is transparent, so we can obtain the transaction information of all the sensors and the relationships between transactions from the DAG network. Thus we can obtain the weights of transactions w and malicious behaviours records $\alpha(\mathcal{B})$ by sweeping the DAG structure easily.

After we calculate Cr_i^P and Cr_i^N respectively, we can get Cr_i according to Eqn. 2. Similar to the definition of the difficulty of mining in Bitcoin [15], the difficulty of PoW in this system is inversely proportional to the credit scores, which is adjusted dynamically throughout the lifetime of the system. We define $Cr_i = \delta \frac{1}{D_i}$, where D_i is the difficulty of PoW for node i, δ is a scale factor ($\delta = 11$ in this paper). So, there is still a question, how to control the difficulty of PoW algorithm?

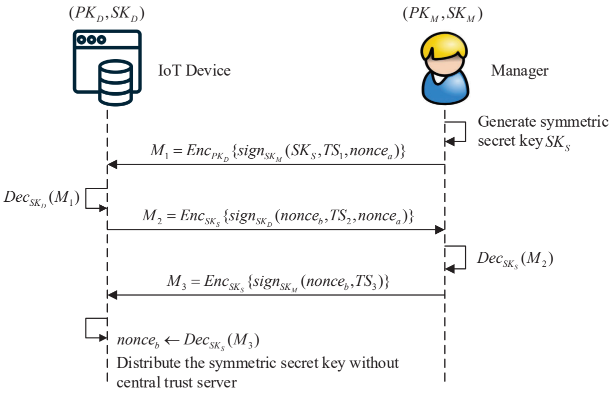

(PK_D, SK_D)
IoT Device
(PK_M, SK_M)
Manager
Generate symmetric secret key SK_S
$M_1 = Enc_{PK_D}\{sign_{SK_M}(SK_S, TS_1, nonce_a)\}$
$Dec_{SK_D}(M_1)$
$M_2 = Enc_{SK_S}\{sign_{SK_D}(nonce_b, TS_2, nonce_a)\}$
$Dec_{SK_S}(M_2)$
$M_3 = Enc_{SK_S}\{sign_{SK_M}(nonce_b, TS_3)\}$
$nonce_b \leftarrow Dec_{SK_S}(M_3)$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

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

In tangle, a new transaction should 'bundle' with two former transactions through PoW algorithm before submitting, which can be expressed in formula 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.

Due to the computing complexity and anti-collision of hash algorithm, we know that if the demand of minimum length of prefix zero is bigger, 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.

Hence, credit-based PoW mechanism can decrease the power consumption of honest nodes while defending 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 the benefit 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 the 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 received message is not tampered or damaged. TS in each message presents timeliness of the message, which is used to resist the 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.

Because the function of each device is relatively fixed, hence, for those devices whose collected data is not sensitive, they do not need to encrypt sensor data. So 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.

IV. Implementation

We implement the proposed system in order to conduct evaluation and analysis on it. In this section, we present the detailed implementation of our system. We implement our system by modifying IOTA implementation, 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, which are manager and gateway. They are implemented based on IRI[2], 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, We modify IRI to provide the credit-based PoW mechanism and integrate the functionality of symmetric key generation and distribution into full nodes, we use the SHA-256 algorithm to distribute secret keys, and use the AES block cipher algorithm implemented by C to encrypt sensor data.

[2] [Online]. Available: https://github.com/iotaledger/iri

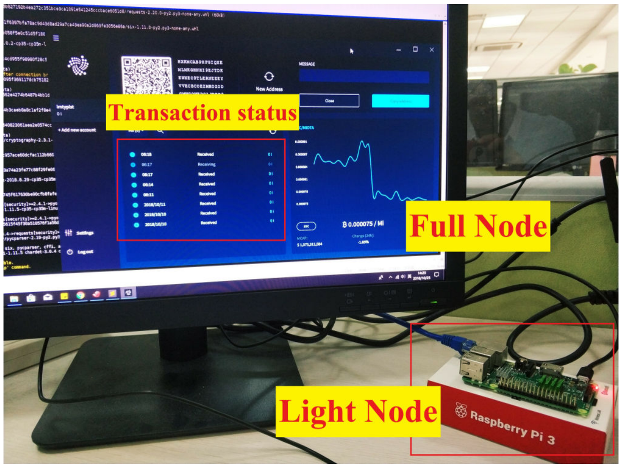


Fig. 5. The implementation of system on PC and Raspberry Pi.

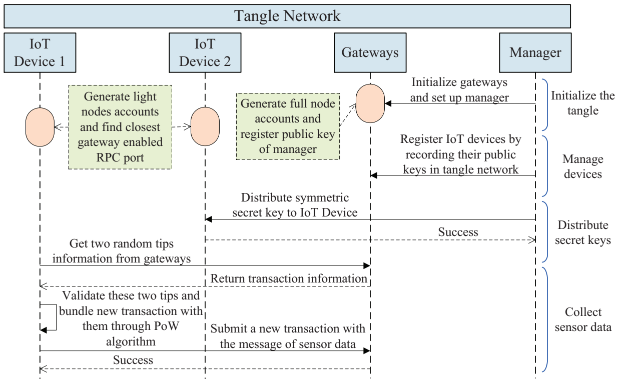

Tangle Network
IoT Device 1
IoT Device 2
Gateways
Manager
Generate light nodes accounts and find closest gateway enabled RPC port
Generate full node accounts and register public key of manager
Initialize gateways and set up manager
Initialize the tangle
Register IoT devices by recording their public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

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

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[3], 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 implementation specification of package is based on aforementioned design of credit-based PoW mechanism. We also implement the AES-based data authority management method on light nodes by using C to encrypt collected sensor data.

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 blockchain.
3) In the stage of secret key distribution, the manager does not need to distribute the 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.
5) When validation is passed, the IoT device bundles the new transaction with these two verified tips through the PoW algorithm, and submits it to the gateways.

Step 4 and step 5 are a single process for sensor data submission, which can be done repeatedly.

V. Evaluation and Analysis

In this section, we evaluate performance in credit-based PoW mechanism and how the introduction of data authority management impact on the efficiency of transactions. In addition, we provide security analysis of the whole system from two aspects, i.e., system security and privacy. IOTA already provides official live transaction visualizer[4], which also displays the 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 Industrial IoT devices, in order to be closer to the actual application scenario, all experiments were done on a Raspberry Pi Model 3B with Quad Core@1.2GHz, which is a power-constrained and computing-limited device.

A. Performance in Credit-Based PoW Mechanism

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

We run PoW algorithm with increasing difficulty to find the relationship between running time and difficulty of PoW, the result is shown in Fig. 7.

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

[3] [Online]. Available: https://pyota.readthedocs.io/

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

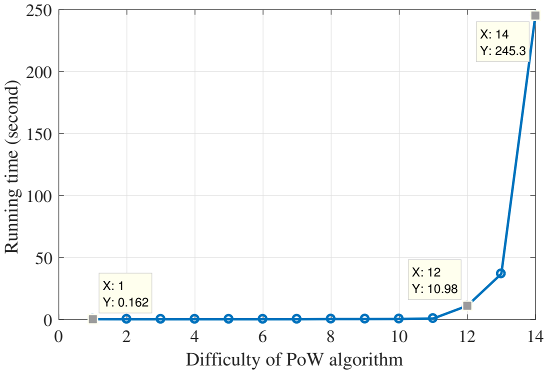


Fig. 7. Running time of PoW algorithm with increasing difficulty.

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 is from 1 to 14, and set 11 as the initial difficulty of PoW, which is the relatively appropriate initial value.

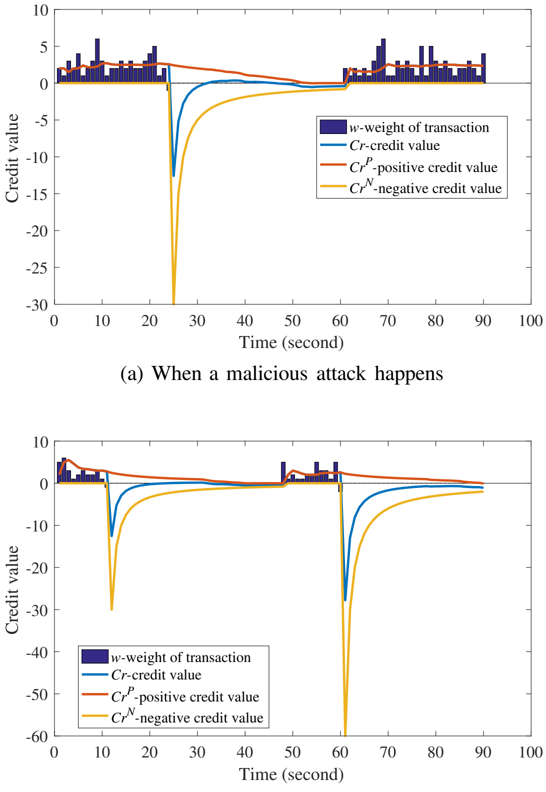
In addition, according to Eqn. 2, there are four tunable parameters, which are λ_1, λ_2, ΔT, $\alpha(\mathcal{B})$. The weight of each transaction w can be counted from tangle network. Here we set $\lambda_1 = 1$, $\lambda_2 = 0.5$. According to the Eqn. 4, the negative part of credit scores has a greater gain, so we set it to 0.5. If you want a more severe punishment mechanism, you can set it bigger. Considering the frequent requests in IIoT systems, we set $\Delta T = 30$ seconds, a not so long interval. And we set $\alpha(\mathcal{B}) = 0.5$ for event $\mathcal{B}$ is lazy tip and $\alpha(\mathcal{B}) = 1$ for event $\mathcal{B}$ is double-spending. From the definition of these two abnormal behaviours in Section III-B, we know that double-spending will cause the rollback of transactions, which impacts the system much more severe than lazy tips. Thus, we set double-spending behaviours to 1. Of course, they can be set to other value if needed because they are adjustable parameters.

We simulate behaviours of a light node to present working mechanism of credit-based PoW, which is shown in Fig. 8.

The x-axis represents the time sequence, 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 also denotes weight of transactions for bars, especially, 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 $Cr^N = 0$, which means the node does not conduct any malicious behaviour before, so the negative credit part is 0. Once the node does any abnormal behaviour, it will be detected immediately and there will be the 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 a 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, so that the node has to take a long 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, so 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. With time goes, the credit value of node will rise gradually and return to normal transaction rate. Besides that, we can notice that, in Fig. 8 (b), if the node conducts malicious attacks twice or more, it will take a longer time to recover normal transaction rate, which can well prevent malicious nodes from attacking. The simulation results indicate that credit-based PoW mechanism can defend malicious attacks efficiently.



(a) When a malicious attack happens

(b) When two malicious attacks happen

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

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

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

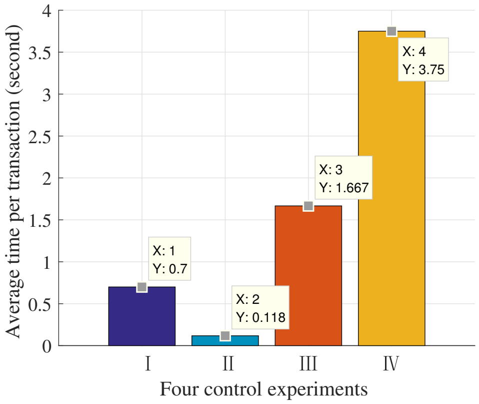


Fig. 9. Performance evaluation in credit-based PoW mechanism. The four control experiments respectively represent *original PoW*, *credit-based PoW with normal behaviours*, *credit-based PoW with a malicious attack*, *credit-based PoW with two malicious attacks*.

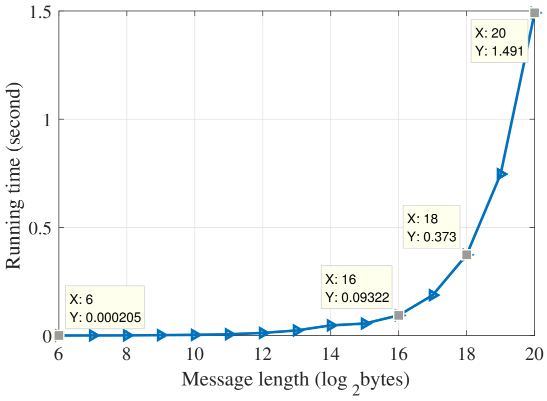


Fig. 10. Impact of symmetric encryption algorithm on transaction efficiency.

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

Due to the introduction of data authority management method in our system, so we evaluate this method's influence level on transaction efficiency. In the method, there mainly contains two components, which are the secret key distribution and sensor data encryption. Consider the frequency of use, key distribution will not be conducted frequently, even only will be conducted once at the initialization of system, whose impact on transaction can be ignored. Thus, in this part, we focus on evaluating performance in sensor data encryption.

As introduced in Section IV, we adopt 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 big 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 introduction of data authority management method has little impact on transaction efficiency.

C. Security Analysis

In this part, we firstly present several possible attack models, and then analyze security from two aspects, i.e., system security and privacy.

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

- **Single Point of Failure**. A single point of failure is a part of system that, if it fails, will stop 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** and **Double-spending**. These two micro possible attacks have already been introduced in Section III-B.

These four possible attacks can be divided into macro attacks and micro attacks. We firstly analyze two possible macro attacks, i.e., single point of failure and Sybil attack.

Our system 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. Also, we know that information recorded in blockchain cannot be tampered, so we can leverage this feature to manage IoT devices and refuse to provide services for unauthorized IoT devices, which can effectively defend attacks like DDoS, Sybil attack.

For two micro possible attacks, i.e., lazy tips and double-spending, the proposed credit-based PoW mechanism in this work also helps to punish and defend malicious nodes, which is presented before. Besides that, consensus mechanisms in blockchain can prevent double-spending effectively. These mechanisms guarantee system safety in the blockchain-based IIoT system.

To be noticed that, in the proposed system, the duty of the manager is to authorize and manage IoT devices. And, the manager is usually the shareholder of the IIoT system, who is the biggest beneficiary in this system. Thus, the manager has no motivation for evil, and it is always considered as an honest node in this system. Under this premise, light nodes and other full nodes are authorized and managed by the manager, which can improve credibility to a certain extent. Even though these devices changes to malicious nodes suddenly, our proposed credit-based PoW mechanism can prevent malicious behaviours efficiently, which has been demonstrated in Fig. 8.

In protecting data privacy, due to the transparency of blockchain, we utilize the symmetric encryption algorithm to implement a data authority management method, which protects sensor data confidentiality through encrypting data before storing in blockchain. Only people who have the secret key can decrypt and get sensor data, which realizes data authority control in a transparent system. Also, the introduction of this method brings little impact on transaction efficiency, which is resource-friendly to IoT devices.

VI. Related Work

In industrial IoT system, there are common technical challenges [1], [16] needed to tackle such as scalability, dependability, privacy, access control, etc. In this section, we review related work carried out for solving these challenges and discuss the insufficiencies of them briefly.

There are some existing solutions that are not based on blockchain technologies. As an example, C. E. Kaed et al. [17] present a semantic rules engine for industrial 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. [18] present a HealthIIoT-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.

There are also many research combining blockchain with IoT to solve the aforementioned issues. For example, A. Dorri et al. [3] propose a Blockchain-based smart home framework to achieve security goals of confidentiality, integrity and availability. But they eliminate the concept of PoW to speed up efficiency of transactions, which will raise system security risks. Also, K. Christidis et al. [19] adopt a similar implementation, i.e., use a white-list scheme, to cancel consensus mechanisms in private networks, thus it faces the same secure issues. Z. Shae et al. [20] propose a blockchain platform for clinical trial and precision medicine, which still stuck in the concept stage and is a lack of evaluation. K. R. ?zy?lmaz et al. [21] try to integrate low-power IoT devices to a blockchain-based infrastructure, but the system is implemented on Ethereum blockchain, which is overload for IoT devices. And the low throughput of Ethereum blockchain cannot satisfy the demands of IIoT system. Di Pietro et al. [22] describe 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 simply apply blockchain technology into IoT system and do not present a detailed implementation.

VII. Conclusion

In this work, we propose a blockchain-based IIoT system in the applied scenarios of smart factory to address aforementioned challenges for IIoT. 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 IIoT systems. Also, the data authority management method can protect data privacy without affecting the system performance, which is also practical in IIoT system. The results of extensive experiments and evaluation show that our system has a good performance in IIoT.

This work will be of importance to research in distributed industrial IoT systems by providing a practical DAG structured blockchain based solution. Our solution is not only suitable for smart factory, but also able to adapt to various IIoT scenarios. However, there are still some limitations in our systems, 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 work was supported in part by the National Key R&D Program of China 2018YFB1004703, China NSF grant 61672349, 61672353, 61373155, and CCF-IFAA Collaborative Research Project Z50201800178.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 33–39, August 2018.

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

[9] 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.

[10] 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.

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

[12] R. B?hme, 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.

[13] A. Churyumov, "Byteball: A decentralized system for storage and transfer of value," *URL https://byteball. org/Byteball. pdf*, 2016.

[14] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng, "Survey on blockchain for internet of things," *Computer Communications*, 2019.

[15] M. Vukolić, “The quest for scalable blockchain fabric: Proof-of-work vs. bft replication,” in *Open Problems in Network Security*. Springer International Publishing, 2016, pp. 112–125.

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

[17] 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.

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

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

[20] 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.

[21] K. R. ?zy?lmaz 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.

[22] 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.

Junqin Huang received the B.E. degree in computer science and technology from University of Electronic Science and Technology of China, Chengdu, China, in 2018.

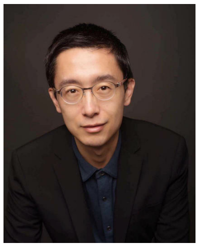
He is currently working toward the Master’s degree in computer technology at Shanghai Jiao Tong University, Shanghai, China. His research interests include big data, Internet of things, blockchain, crowdsensing.

Linghe Kong is currently a tenure-track research professor at Department of Computer Science and Engineering, Shanghai Jiao Tong University. He previously served as a postdoctoral researcher at Columbia University, McGill University, and Singapore University of Technology and Design. He received his Ph.D. degree at Shanghai Jiao Tong University, 2012, his Master degree at TELECOM SudParis (ex. INT), 2007, and his Bachelor degree at Xidian University, 2005. His research interests include wireless networks, big data, mobile computing, Internet of things, and smart energy systems.

Guihai Chen received the B.E. degree in computer software from Nanjing University, Nanjing, China, in 1984, the M.E. degree in computer science from Southeast University, Nanjing, China, in 1987, and the Ph.D. degree in computer science from the University of Hong Kong, Hong Kong, in 1997.

He is a Distinguished Professor at Shanghai Jiaotong University, Shanghai, China. He has been invited as a Visiting Professor by many universities, including Kyushu Institute of Technology, Japan, in 1998, University of Queensland, Australia, in 2000, and Wayne State University, USA, during September 2001 to August 2003. He has a wide range of research interests with focus on sensor networks, peer-topeer computing, high-performance computer architecture, and combinatorics.

Min-You Wu received the Ph.D. degree from Santa Clara University, Santa Clara, CA in 1984. He is a professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University and a research professor with the University of New Mexico. His research interests include grid computing, wireless networks, sensor networks, multimedia networking, parallel and distributed systems, and compilers for parallel computers.



Xue Liu received the B.S. degree in mathematics and the MS degree in automatic control both from Tsinghua University, China, and the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2006. He is a full professor in the School of Computer Science at McGill University. He has also worked as the Samuel R. Thompson associate professor in the University of Nebraska-Lincoln and HP Labs in Palo Alto, California. His research interests include computer networks and communications, smart grid, real-time and embedded systems, cyber-physical systems, data centers, and software reliability. His research interests include computer networks and communications, smart grid, real-time and embedded systems, cyberphysical systems, data centers, and software reliability.

Peng Zeng received the B.S. degree in computer science from Shandong University, Shandong, China, in 1998, and the Ph.D. degree in mechatronic engineering from Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2005. From 2005 to 2007, he was an Associate Professor with Shenyang Institute of Automation, Chinese Academy of Sciences, where he was involved in research on wireless sensor networks.

He is currently a Professor at Shenyang Institute of Automation, Chinese Academy of Sciences. His current research interests include wireless sensor networks for industrial automation, smart grids, and demand response.