IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 28, 2019, accepted March 14, 2019, date of publication March 18, 2019, date of current version April 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2905846

A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT

SHENG DING[1], JIN CAO[1], CHEN LI[2], KAI FAN[3], AND HUI LI[1]
[1]School of Cyber Engineering, Xidian University, Xi'an 710071, China
[2]School of Telecommunications Engineering, Xidian University, Xi'an 710071, China
[3]State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an 710071, China

Corresponding author: Sheng Ding (shawnding.xdu@gmail.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802700 and in part by the National Natural Science Foundation of China under Grant 61772404 and Grant U1836203.

ABSTRACT With the sharp increase in the number of intelligent devices, the Internet of Things (IoT) has gained more and more attention and rapid development in recent years. It effectively integrates the physical world with the Internet over existing network infrastructure to facilitate sharing data among intelligent devices. However, its complex and large-scale network structure brings new security risks and challenges to IoT systems. To ensure the security of data, traditional access control technologies are not suitable to be directly used for implementing access control in IoT systems because of their complicated access management and the lack of credibility due to centralization. In this paper, we proposed a novel attribute-based access control scheme for IoT systems, which simplifies greatly the access management. We use blockchain technology to record the distribution of attributes in order to avoid single point failure and data tampering. The access control process has also been optimized to meet the need for high efficiency and lightweight calculation for IoT devices. The security and performance analysis show that our scheme could effectively resist multiple attacks and be efficiently implemented in IoT systems.

INDEX TERMS Access control, attribute-based access control, blockchain, consortium blockchain, Internet of Things.

I. INTRODUCTION

There is no doubt that the Internet of Things (IoT) is one of the most promising technologies and has attracted broad attention from academia and industry in recent years. IoT is a novel architectural framework integrating the physical world with the Internet over existing network infrastructure. It aims to connect all of the intelligent devices, including physical devices, vehicles and home appliances, and enable them to collect and share data autonomously through the Internet. According to the forecast from Gartner,[1] more than 8.4 billion connected things joined this network worldwide in 2017, up 31 percent from 2016, and will reach 20.4 billion by 2020. However, the sharp increase in the number of connected devices brings new security risks and challenges to the IoT systems. As IoT devices are widely distributed, it is so difficult to enforce strict security control that makes them vulnerable to various attacks by malicious adversaries. It is necessary to protect IoT devices from unauthorized access which will usually lead to severe data leakage, as these devices often contain much valuable and sensitive data. As we know, access control is one of the most important technologies for guaranteeing the security data. Traditional access control technology such as discretionary access control (DAC), identity-based access control (IBAC), are not suitable for implementing access control in IoT systems, because it is almost impossible to make an access control list (ACL) for everyone in the IoT system on account of the huge quantity of unknown identities. Another common technique mandatory access control (MAC) is generally enforced by a central administrator, which exists the problem of single-point failure. As IoT devices may belong to different management organizations due to whether their location or function, centralized access control mode does not fit for IoT systems.

Attribute-based access control (ABAC) provides a type of flexible, dynamic and fine-grained access control. It abstracts the roles or the identities into a set of attributes issued by the

The associate editor coordinating the review of this manuscript and approving it for publication was Kuan Zhang.

[1] https://www.gartner.com/newsroom/id/3598917

attribute authorities. An access policy described by a Boolean formula over a set of attributes is used to define the valid and authorized access. There is no longer need to assign roles or make access control lists for each one in the system. Instead, the attribute authorities only need to manage each attribute defined in the system and distribute them to proper users. In this way, access management can be effectively simplified as the number of attributes is much less than the number of users in the system.

Blockchain is another hot topic of interest among technology giants and business communities. It is an open, transparent and distributed ledger that record transactions between two parties efficiently in a verifiable and permanent way [1]. Once recorded, the data on the blockchain can not be tampered unless a new consensus is reached. Combining IoT with blockchain technology is a promising trend and is expected to ensure trust and reduce overall overhead for IoT systems. It can help IoT to establish a decentralized, credible and publicly verifiable database so that billion of connected things can achieve a distributed trust through it.

In this paper, we proposed a novel attribute-based access control scheme using blockchain for IoT systems. Our main contributions are summarized as follows:

1) We proposed a novel attribute-based access control scheme for IoT systems. There is no longer need to make ACL or assign roles for everyone in the system. Each device can be described by a set of attributes which are predefined in the system and issued by the attribute authorities according to its identity or ability. No one is allowed access unless it has enough attributes that match the access policy.
2) We used blockchain to record the distribution of attributes. The attributes authorities jointly maintain a public and credible ledger of ''transactions''. Once recorded, the data in the block can not be altered and anyone can inquire the blockchain at any time when needed.
3) We simplified the access control protocol and the two parties involved only need to do some simple signature and hash operations. In this way, our scheme becomes more effective for the devices with limited computing capability and energy supply in IoT systems.

This manuscript is organized as follows: related work is summarized in Section 2, followed by preliminaries in Section 3. We propose the detailed construction of our attribute-based access control scheme using blockchain for IoT systems in Section 4. Section 5 and section 6 are the security and performance analysis respectively. Finally, we end up with a conclusion in Section 7.

II. RELATED WORK

To ensure strict access control, authentication is a necessary mechanism to confirm the identity of the participant involved in the communication before sharing or exchanging data with each other. Salman *et al.* [2] proposed an identity-based authentication scheme for IoT. In their scheme, each device is managed by the gateway it belongs to and has a virtual IPv6 address granted by a controller as its unique identity. The authorized address can be used as a certificate when authenticating with others. Porambage *et al.* [3] designed a lightweight authentication mechanism for wireless sensor networks in distributed IoT applications. Each sensor node will first be issued a credential from corresponding cluster head as a prerequisite for future authentication. And four types of communication links are discussed for the difference in the geographic distribution of nodes and the relationship between them. Shivraj *et al.* [4] used the one time password (OTP) technique to designed a lightweight end-to-end authentication scheme based on identity-based elliptic curve cryptosystem (IBE-ECC). As there is no need to store the keys and the key size is small, the proposed scheme turns out to be efficient for IoT systems. Reference [5] proposed a distributed capability-based access control scheme for IoT system, in which smart things could realize a lightweight end-to-end authorization. A capability is generally a communicable and unforgeable token of authority.

Many enterprises are inclined to act as centralized trusted authorities and enforce authorization based on OAuth protocol [6]. The project Connect All IP-based Smart Objects (CALIPSO)[2] implements a centralized system model, in which smart objects will delegate the authorization to a powerful server named IoT-OAS. However, [7] showed that run all OAuth logic in a resource-constrained device is almost impossible because of its high communication and computation overheads.

Role-based access control (RBAC) [8] is a common approach to restrict the access privilege to authorized users. It grants the specific privileges to users according to their roles in the system. However, it is unsuitable for IoT systems, as this kind of access model is not flexible and scalable enough. Once a device has been assigned to a role, it could only access data in a fixed manner. Compared with RBAC, attribute-based access control (ABAC) [9] is more flexible and scalable, and could provide a more fine-grained access control. An entity could define a access policy over attributes to restrict the valid access. For perception layer of IoT, [10] presented an efficient access control scheme based on ABAC model. Only one with a satisfied subset of attributes in the policy can get the access authorization. Zhang *et al.* [11], [12] proposed two attribute-based encryption schemes to ensure fine-grained access control as well as data security.

Most of the access control schemes above face a common problem that a credible center is needed for ensuring trust. As IoT devices are worldwide distributed, there can not be a centralized node to manage all of them. Generally, each of them is managed by the authorities nearby or which it belongs to. To build trust among these authorities is the foundation of implementing distributed management. We therefore think

[2] https://cordis.europa.eu/docs/projects/cnect/9/288879/080/deliverables/001-D553.pdf

of using blockchain to establish a distributed consistent trust. It was first presented by Nakamoto [13] in 2008, which is now the underlying technology of digital currency Bitcoin.

Ouaddah *et al.* [14] presented an extensive review of different access control schemes for IoT and the conclusion is that traditional access control mechanisms are not appropriate for resource-constrained environments. Hardjono and Pentland [15] described the design of an access control and identity management based on blockchain in detail. The ChainAchor system they proposed in [15] provides anonymous but verifiable identities for entities in the system. Consensus nodes could refer to a list of anonymous members' public keys to enforce access control. The transactions made by an same entity are unlinkable. Hardt et al. [16] systematically summarized the implementation of blockchain in IoT.

In Ouaddah [17] introduced a decentralized authorization management framework using blockchain named FairAccess. It uses new types of transactions to grant or revoke access for users. Novo [18] proposed a fully distributed access control scheme for IoT systems based on blockchain. Unlike FairAccess, the policy rules of the management system are defined through generating a single smart contract and the access control policies are defined by creating transactions toward this smart contract in [18]. We use consortium blockchain instead of the private blockchain to establish a more decentralized system. The access policies made up of attributes are defined by the IoT devices themselves according to their security requirements. To get the access authorization, the device involved must prove its ownership of corresponding attributes which satisfy the policy.

III. PRELIMINARIES

A. CONSORTIUM BLOCKCHAIN

Generally speaking, there are three categories of blockchain: public blockchain, consortium blockchain and private blockchain. Unlike public or private blockchain, the verification of transactions is executed by a set of pre-selected nodes in consortium blockchain. In other words, these nodes jointly maintain the public ledge, neither all nodes together nor a certain node with full control. It is partially decentralized and provides a hybrid between the low-trust of public blockchain and the single highly-trusted entity model of private blockchain.[3] Compared with public blockchain, consortium blockchain replaces costly Proof of Work (PoW) with other efficient consensus algorithm, such as Practical Byzantine Fault Tolerance (PBFT) [19] and Raft [20]. There is no need for consortium blockchain to wait for the validation of at least six blocks just as in Bitcoin which uses POW as its consensus algorithm. Therefore, consortium blockchain takes much less time to reach a consensus than public blockchain. It makes consortium blockchain more suitable for IoT, which is more time-sensitive and needs a high efficiency of communication. In principle, once the data is recorded in the blockchain, no one could tamper it unless more than 51 percent of all the nodes reach a new consensus in the public blockchain with PoW consensus algorithm. But for consortium blockchain, if errors occurred, the consortium nodes could timely fix them through manual intervention in a short time. In our scheme, the attribute authorities who take charge of the attribute distribution play the role of these nodes. If one of them generates a transaction of attributes, the transaction will be first put into its transaction pool. Once a leader is elected, it will pack the transactions into a block and publish the block to the others for reaching a consensus. Once reached, the new block will be appended to the end of the blockchain.

B. BLOCK STRUCTURE

Blockchain is defined as a series of blocks connected by hash. Each block is divided into two parts: block header and block body. All the transactions involved in a block make up the block body. The block header consists of a hash of the last block header defined as Prev_hash, a timestamp and a Merkle root of the transaction data. These blocks connect one by one and finally form a chain, as described in Fig.1. The Merkle root is used to efficiently check the integrity of the transaction data. The timestamp is added for showing when the block is generated and making all of the blocks sorted by time. The hash of the last block header contains all of the information about the last block, which ensures the integrity of the block data. If some transactions in the previous block are maliciously altered, the Merkle root of all the transactions involved in that block will be also changed, which results in a change of the hash of its block header. This change will iteratively spread to the subsequent blocks, and finally forms a fork. However, this new chain will not be a consensus which all of the consortium nodes agree on. Hence, blockchain is inherently resistant to data tampering with this ingenious kind of structure. In our scheme, each block body is filled with transactions of attributes packed by a leader node which is elected by the other consortium nodes. A transaction of an attribute represents an authorization of this attribute from an attribute authority to a provided address.

C. ADDRESS AND TRANSACTION

In blockchain technology, the term "address" is represented as a string of characters which consists of digits and letters,

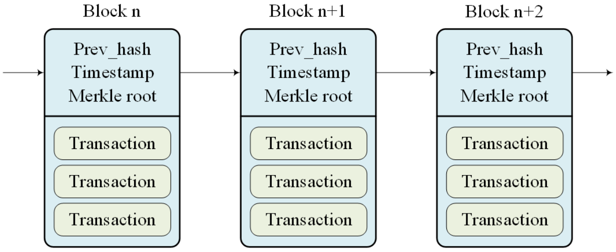


FIGURE 1. The structure of a blockchain.

[3] https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/

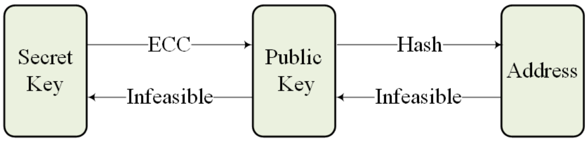


FIGURE 2. Generate an address.

e.g., "1EyXAQaN5XhEtkWaBhAMYACHBSADmn5z8". Each device has a series of public and secret key pairs. The secret key is randomly chose by the device itself based on elliptic curve cryptography (ECC) and its corresponding public key is generated by multiplying the generator G of the elliptic curve which is defined by the system. With this public key, the address can be calculated through hash function and Base58Check encoding. Given only a public key, it is almost infeasible to find its secret key pair in polynomial time, which is guaranteed by the elliptic curve discrete logarithm problem (ECDLP). And it is also difficult to retrieve the public key from the address, which is ensured by the non-invertibility of hash function, as shown in Fig.2.

The secret key acts as a certificate of the ownership of the corresponding address. If a signature generated by a secret key and a public key are presented to prove its ownership of an address, the verifier can check whether the public key could not only verify the signature but also generate the address. In our scheme, the address is used to apply for attributes. After verifying the identity of the applicant, the attribute authority will distribute proper attribute which is requested towards the provided address, each such transaction is in the form of

$$AA \xrightarrow{i} Address,$$

where AA is the address of the attribute authority, i is the requested attribute and $Address$ is the provided address.

IV. PROPOSED SCHEME

A. SYSTEM MODEL

We first present the system model of our access control scheme using blockchain for IoT, as described in Fig.3 There are two main entities, attribute authorities and IoT devices. The attribute authorities act as the consortium nodes in consortium blockchain and the key generation center (KGC) at the same time. We allow at most $(n-1)/3$ out of n attribute authorities to be Byzantine nodes, so there are at least 4 consortium nodes. To facilitate understanding, we only use the minimum number of consortium nodes to construct the system model in Fig.3.

1) attribute authorities

The attribute authorities are the managers of the blockchain and the distributors of attributes. To jointly maintain a distributed ledger, they need a consensus on the attribute distribution. The authorization of the attributes is recorded in the form of transactions. A transaction of attribute authorized by a certain attribute authority is first placed into its own transaction pool. The other attribute authorities must verify its validity before recording into the blockchain. Once successfully recorded, no one can tamper the block data, unless all of the consortium nodes reach a new consensus and generate a new chain from the block which needs to be modified. Considering geographical distribution factors, each attribute authority manages different regions of devices.

At the same time, the attribute authorities also act as the key generation center (KGC) when IoT devices register with the system. Each attribute authority will issue a pair of public and secret key to each affiliated device according to its identity using identity-based cryptography. With the public and secret key pair, the devices involved in a communication could mutually authenticate with each other and agree on a session key.

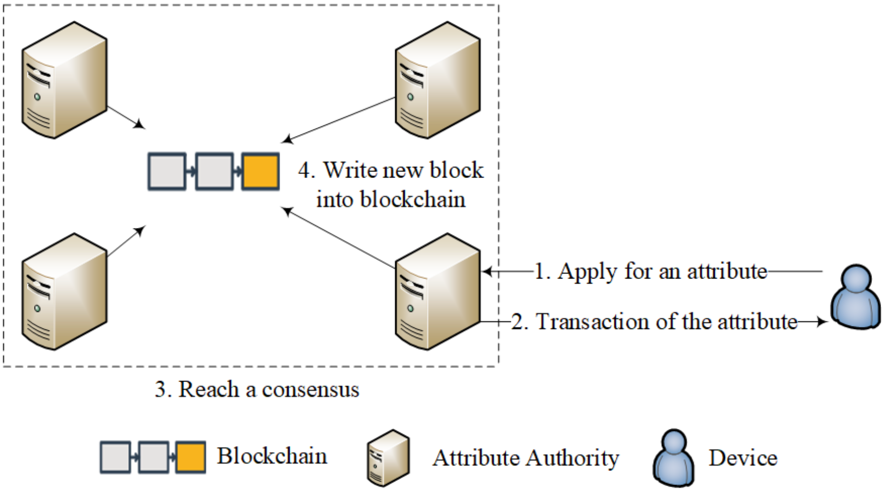

4. Write new block into blockchain
1. Apply for an attribute
2. Transaction of the attribute
3. Reach a consensus
Blockchain
Attribute Authority
Device

FIGURE 3. System model.

2) IoT Devices

The devices are responsible for collecting, processing and sharing the data in IoT systems. They are not involved in the verification of transactions and only have the read permission of the blockchain. To ensure valid access and data security, the data requester need to obtain the access authorization from the data owner before exchanging data. The data requester uses the attributes assigned by the attribute authorities to prove they have corresponding permissions. Only if there is a satisfied set of attributes that matches the access policy of the data owner can the access be allowed.

B. SECURITY MODEL

In our scheme, the attribute authorities may be so vulnerable due to various malicious attacks that become Byzantine nodes. We restrict the number of Byzantine nodes to be no more than $\frac{n-1}{3}$ out of n nodes. With this restriction, the consensus of attribute distribution could be normally reached. The secret key of each attribute authority is securely kept, therefore no one could forge the signature of each consortium node. And they know the public key of each other, so that they could verify the validity of each signature.

The devices are untrusted as they may collude with each other, driven by interests, to authenticate with others when none of them independently has a satisfied set of attributes. Malicious devices may even intend to tamper the blockchain or interfere with attribute authorities to reach a consensus.

C. CONSTRUCTION

1) SYSTEM INITIALIZATION

Let λ be the security parameter. The system initialization algorithm takes in the security parameter λ as input and outputs the global parameters for the system. All of the system members need to agree on a same elliptic curve which is stipulated by the system. Let E be an elliptic curve additive group in which the elliptic curve discrete logarithm problem (ECDLP) is intractable. Let G be an element of E with a large prime order r. Then two secure hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_r^*$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ are also chose to map arbitrary size of bit string into a new bit string of fixed size. Let G be an element of E with a large prime order r. All of the attribute authorities share a $SK_m \in \mathbb{Z}_r^*$ and keep it as the master private key, the corresponding master public key PK_m is therefore SK_mG. Finally the system parameters are published as $\{\lambda, E, G, PK_m, H_1, H_2\}$.

2) REGISTRATION

Each device has an intuitive identity ID in the system. For registration, the attribute authority which the device belongs to will issue a pair of public and secret key to it in a secure channel based on identity-based cryptography upon verification of its identity.

3) ADDRESS CREATION

Each device uses an address along with its ID to apply for an attribute i. To generate an address, it randomly choose a $k \in \mathbb{Z}_r^*$ as the secret key (SK) and hence kG is the public key (PK). To generate a corresponding address, the device can hash $PK\|ID$ ($\|$ denotes concatenation here) and encode the result by Base58Check encoding. Therefore, the address is:

$$Address = Base58Check[H_2(PK\|ID)].$$

4) ATTRIBUTE REQUEST

Each attribute authority has a pair of public and secret key. The public key is used to generate its own address AA and the secret key is used to sign the transactions. The attribute authority which the device interacts with will verify whether the applicant is capable of possessing this attribute i. If the device passes the verification, the attribute authority will generates a transaction:

$$AA \xrightarrow{i} Address.$$

Then the attribute authority signs the hash of this transaction and a timestamp with its secret key, that is

$$Sig_{SK}[H_1(AA \xrightarrow{i} Address\|timestamp)].$$

Finally, the attribute authority packs up the transaction, the signature and the timestamp, and put it into its own transaction pool.

These consortium nodes will periodically elect a block maker. Its duty is to pack the transactions in its transaction pool into a block and broadcast it to the other consortium nodes for reaching a consensus. The block maker sorts the transactions according to the timestamp and compute the Merkle root of the selected transactions. The block header consists of a hash of the last block header, a timestamp of the generation of this block and the Merkle root.

The block maker broadcasts this new block to the other consortium nodes using PBFT[5] protocol as described in Fig.5. In the pre-prepare phase, each of the rest of consortium nodes will verify the validity of this new block and broadcast it to the others in the same way. Once receiving $2f$ same blocks, they will broadcast an acknowledgement message to the others in the prepare phase. And if a node receiving $2f + 1$ acknowledgement message, it will append the new block into the blockchain.

5) ACCESS CONTROL

An access control protocol between Alice and Bob is executed as described in Fig.4:

1) Alice first initiates a communication request to Bob with her identity ID_A and use standard identity-based authentication and key agreement (AKA) protocol to generate a session key K with Bob. The security of the subsequent communication between Alice and Bob is ensured by K using any symmetric key algorithm. For simplicity, we only describe the process of message

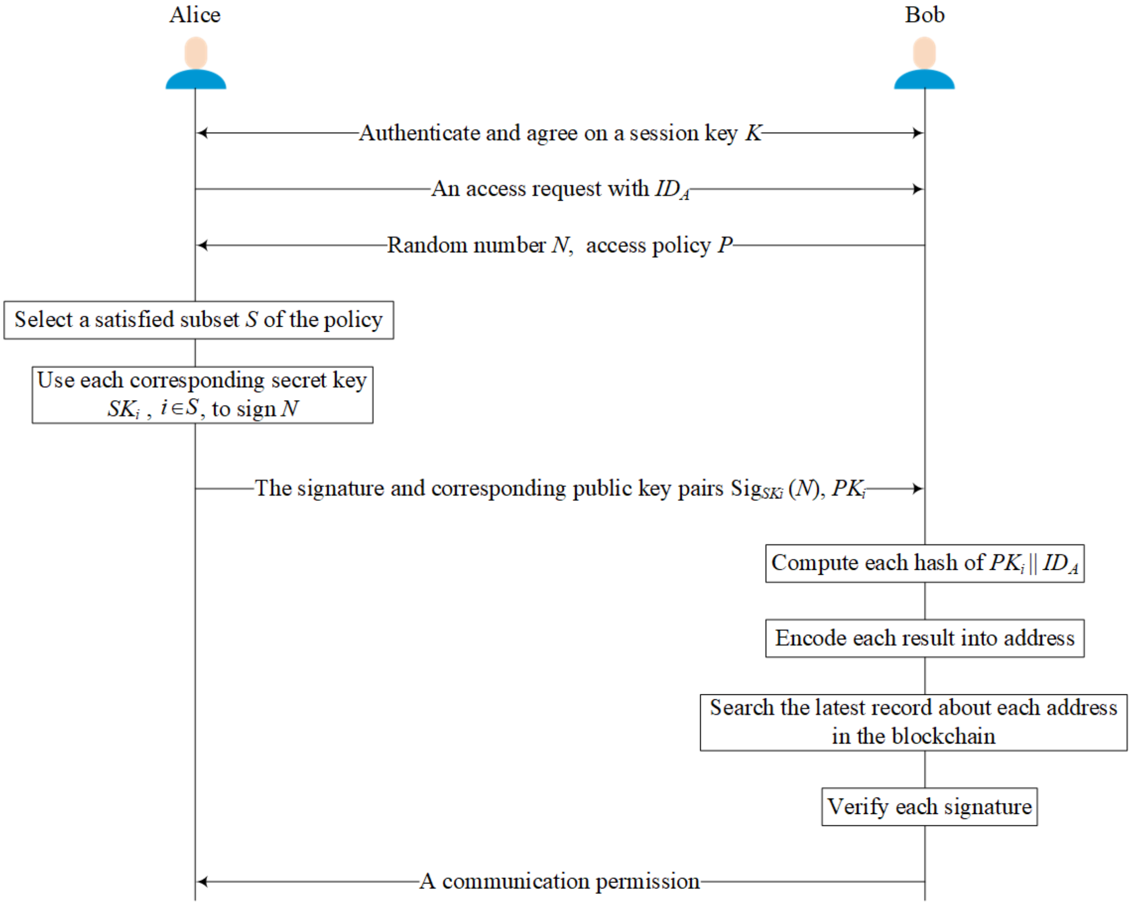

Alice
Bob
Authenticate and agree on a session key K
An access request with ID_A
Random number N, access policy P
Select a satisfied subset S of the policy
Use each corresponding secret key SK_i , $i \in S$, to sign N
The signature and corresponding public key pairs $\mathrm{Sig}_{SKi}(N)$, PK_i
Compute each hash of $PK_i \| ID_A$
Encode each result into address
Search the latest record about each address in the blockchain
Verify each signature
A communication permission

FIGURE 4. Access control between two parties involved in the communication.

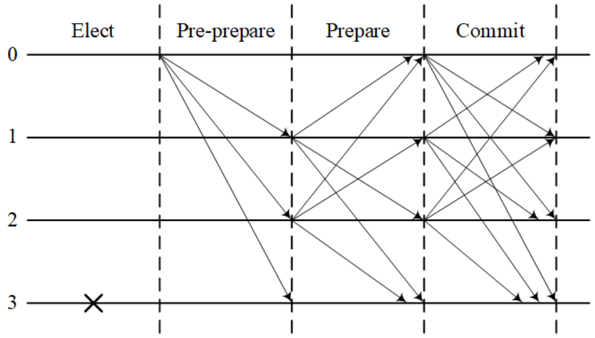


FIGURE 5. Reach a consensus.

exchange below and omit the symmetric encryption for each communication.

2) Then Bob returns a random number $N \in \mathbb{Z}_r^*$ and an access policy P which indicates who can communicate with him.

3) Alice chooses a satisfied subset S of the policy and uses each secret key whose corresponding address has been issued the matched attribute i to sign the random number N. Then Alice returns Bob the satisfied subset of attributes together with each signature and public key pair $Sig_{SK_i}(N)$, PK_i, $i \in S$.

4) Bob first hashes the $PK_i \| ID_A$ and encodes the result by Base58Check encoding to get the corresponding address. Then he searches the blockchain to find out the latest related record about this address. If this address contains the claimed attribute i, Bob use the public key PK_i to verify the signature $Sig_{SK_i}(N)$ by computing:

$$Ver_{PK_i}(Sig_{SK_i}(N)) \overset{?}{=} N.$$

If so, it is demonstrated that Alice really possesses this address and the attribute contained in this address indeed. Finally, Bob checks whether the submitted set of attributes satisfy the access policy he specified.

5) If Alice possesses enough attributes which satisfy Bob's access policy, Bob will allow Alice's request to access her data. The process of the data transmission could be also encrypted using the session key K generated in the first step.

D. REVOCATION

The attribute distribution should be dynamic and scalable for enforcing correct access control. To make attributes characterize identities more accurately, the system must be able to timely revoke the attributes which are expired or not to be owned by a specific user. To revoke an attribute i from a user, the attribute authority could generate a new transaction of this attribute:

$$AA \overset{i}{\leftarrow} Address,$$

and execute the consensus protocol with the other consortium nodes again. The new block which contains this revocation transaction will be appended to the blockchain after they reach a new consensus. When Bob searches the blockchain, the latest related record about the address of that attribute i is its revocation, rather than the previous authorization of it. In this way, an effective and efficient revocation of attributes is realized.

V. SECURITY ANALYSIS

In this section we first theoretically analyze and illustrate that our attribute-based access control scheme is collusion and impersonation resistant. Then we simulate our access control scheme with the formal verification tool Automated Validation of Internet Security Protocols and Applications (AVISPA),[4] which is widely accepted and used for the automatic security analysis of Internet protocols and applications, to prove its security in practical application.

A. COLLUSION RESISTANT

To ensure correct access control, the proposed scheme must be able to resisting collusion attack. Sometimes the devices may collude with each other, driven by interests, to attempt authenticating with others for valuable data when none of them independently has a satisfied set of attributes. It absolutely violates the original intention of enforcing access control for secure communication. Suppose that Bob has specified an access policy $X \wedge (Y \vee Z)$, which means only a device with attributes $X \wedge Y$ or $X \wedge Z$ can communicate with him. Unfortunately, Alice only has the attribute X and another one Eve has only the attribute Y. It's obviously that if Alice colludes with Eve, they two will have a satisfied set of attributes. As there is not a global ID to bind various attributes belonging to a specific user, it is quite difficult for Bob to distinguish whether the submitted attributes are all owned by a same individual. However, in our scheme, to get the address of the claimed attribute i, Bob needs to hash corresponding public key submitted by Alice together with ID_A, and encode the result by Base58Check encoding into the address, that is

$$Address = Base58Check[H_2(PK_i \| ID_A)].$$

Although Eve may give Alice his attribute Y, including an address which contains this attribute, the corresponding public key and sign the random number instead of Alice, the difference in ID can not be changed. If Alice shows Bob the combination of the address, the corresponding public key and the signature of attribute Y according to the protocol, Bob will detect that the address

$$Base58Check[H_2(PK_Y \| ID_A)]$$

is not equal to the address of attribute Y submitted by Alice which is originally

$$Base58Check[H_2(PK_Y \| ID_E)]$$

owned by Eve. In this way, Bob could see through the trick played by the Alice and Eve and terminate the authentication. Hence, our access control scheme is collusion resistant.

B. IMPERSONATION RESISTANT

In traditional PKI-based authentication scheme, digital signature is a credible evidence to prove identity. As the digital signature is produced by the secret key exclusively owned by a specific user, it gives the receiver enough reason to believe the message was sent by the claimed sender and nobody else could forge it. However, in the blockchain scenario, the secret key is generated by the user itself and there is not a certificate authority to issue digital credential for the corresponding public key. As a result, if malicious users intercept a signature, they could impersonate the actual owner of this signature by simultaneously showing the signature and the relevant public key which is easy to obtain. In our scheme, even if Bob reserves the signature signed by Alice, the signature is useless in a new round of access control as each round will generate a new random number N' to be signed. As the secret key is securely kept by each user, malicious users could not forge the signature of others without the relevant secret key. And if Bob replays the signature $Sig_{SK_i}(N)$ signed by Alice, the receiver could easily detect that the random number N in the signature is not the N' generated by him.

Suppose there are actually three parties involved, namely Alice, Bob and Cindy. Bob intends to access Cindy's data, however, Bob does not possess a satisfied set of attributes. In other words, Bob can not provide corresponding signatures to Cindy for demonstrating she has enough attributes. If Bob launch a man-in-the-middle attack, as shown in Fig.6, she could alter her access policy so that the attributes in the new policy match Cindy's policy. At the same time, Alice, with enough attributes, intends to access Bob's data. To get the authorization from Bob, Alice needs to provide corresponding signatures of the random number chosen by Bob. In order to get the signature needed by Cindy, Bob may replay the random number selected by Cindy to Alice. In this way, Bob could get a satisfied set of signatures even though she does not have corresponding attributes. However, if Bob send them to Cindy for asserting that she has enough attributes, Cindy could easily detect that the address generated by hashing and encoding the $PK \| ID_B$ has not been granted corresponding attribute and deny the access request from Bob. Therefore, our access control scheme can resist impersonate attacks.

[4] http://www.avispa-project.org/

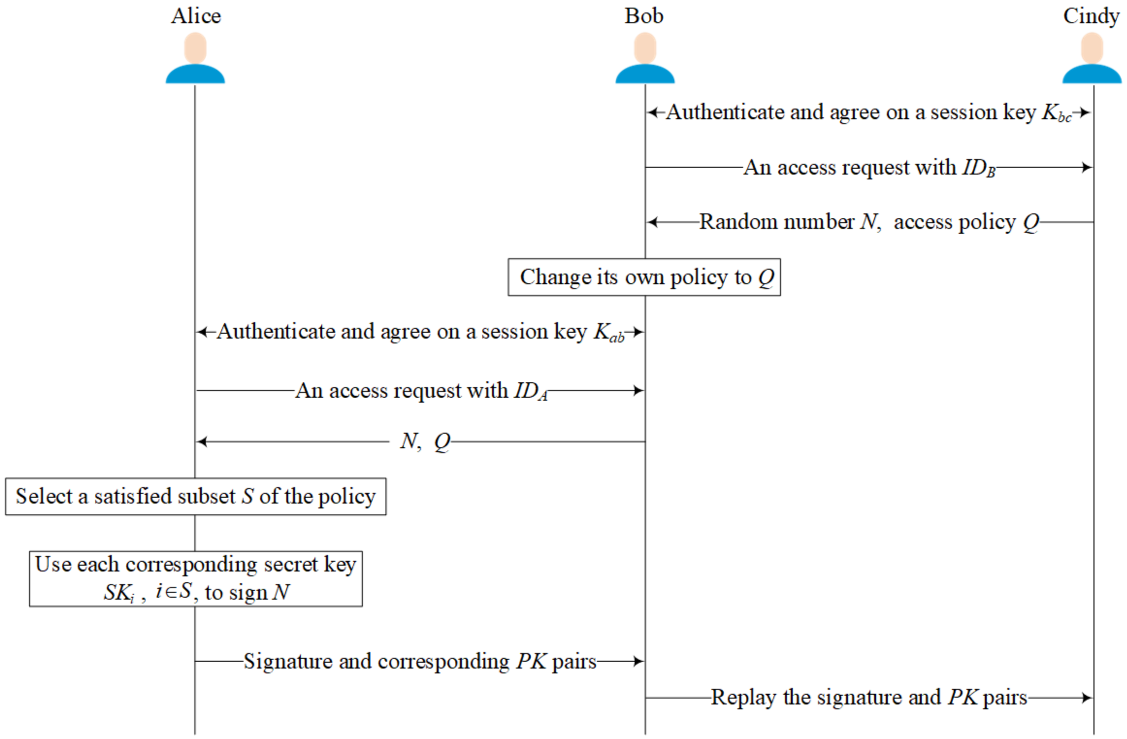
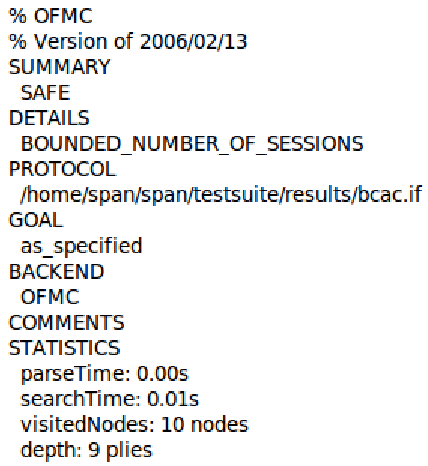

Alice
Bob
Cindy
Authenticate and agree on a session key K_{bc}
An access request with ID_B
Random number N, access policy Q
Change its own policy to Q
Authenticate and agree on a session key K_{ab}
An access request with ID_A
N, Q
Select a satisfied subset S of the policy
Use each corresponding secret key SK_i , $i \in S$, to sign N
Signature and corresponding PK pairs
Replay the signature and PK pairs

FIGURE 6. An MITM attack launched by Bob.

C. AVISPA

The designed protocols and intended security properties need to be first specified in a language called High Level Protocol Specification Language (HLPSL)[5] through the Security Protocol ANimator (SPAN)[6] for AVISPA. Then the AVISPA Tool will translate it into the Intermediate Format (IF)[7] which is a lower-level language than HLPSL by an inbulit translator called hlpsl2if. The IF specification can be directly read by the four back-ends of the AVISPA Tool, namely On-the-fly Model-Checker (OFMC) [21], CL-based Attack Searcher (CL-AtSe) [22], SAT-based Model-Checker (SATMC) [23] and Tree Automata-based Protocol Analyzer (TA4SP) [24]. The AVISPA Tool will output an analysis result based on whether the security goals are satisfied or violated.

Our proposed scheme is simulated using the OFMC back-end with a bounded number of sessions. The intruder model is Dolev-Yao model, under which the intruder could take full control over the network, such that all messages sent by agents will go through the intruder. The intruder may intercept, analyze, or even modify messages as long as he knows the required keys and send them to whoever else in the name of any other agents. The analysis result indicates that our scheme could withstand various attacks such as replay attacks, impersonation attacks and man-in-the middle attacks under the test of AVISPA, and the intended security goal are all satisfied, as shown in Fig.7.



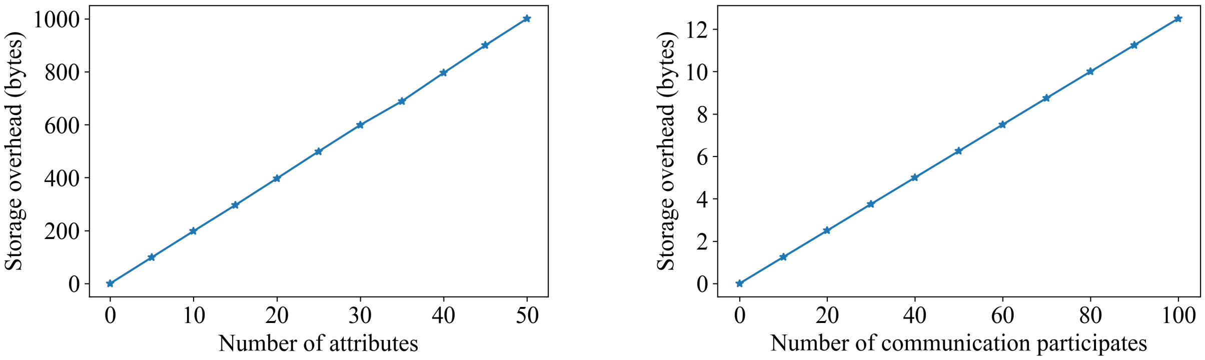
```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/bcac.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 10 nodes
  depth: 9 plies
```

FIGURE 7. Security analysis result using OFMC back-ends.

VI. PERFORMANCE ANALYSIS

We carried out a proof of concept implementation of our attribute-based access control scheme for testing its availability and evaluating its storage and computation overhead. We chose Hyperledger Fabric as the platform to construct the blockchain used in our scheme. It is underpinned by a

[5] http://www.avispa-project.org/delivs/2.1/d2-1.pdf
[6] http://people.irisa.fr/Thomas.Genet/span/
[7] http://www.avispa-project.org/delivs/2.3/d2-3.pdf



(a)

(b)

FIGURE 8. **The storage overhead of session keys and access policy (a) Access policy. (b) Session keys.**

modular architecture delivering high degrees of confidentiality, resiliency, flexibility and scalability. We implement it on an Ubuntu Linux 16.04LTS desktop with Intel Pentium G620 CPU at 2.60GHz and 1GB RAM.

A. STORAGE OVERHEAD

As we know, a majority of devices in IoT are so resource-constrained that the storage overhead is an important factor that must be taken into consideration. It is impractical for these devices to allocate too much storage space for additional data besides the valuable data they have collected. Hence, we analyze the storage overhead and clarify its reasonability for each of them respectively, as shown in Fig.8. There are mainly three kinds of additional data that needs to be stored locally for each device, namely the global parameters, the session keys and the access policy.

- **Global parameters** All of the entities in the IoT system need to share a same set of global parameters. It specifies the security parameter, the elliptic curve, the hash functions involved, the master public key of the attribute authorities and the public and secret key pair of each device itself. With these parameters, our attribute-based access control scheme using blockchain for IoT could be correctly executed. The global parameters are fixed after system initialization and the size of them are obviously acceptable for those resource-constrained devices in IoT.
- **Access policy** The size of the access policy of each device is linear to the number of attributes according to its complexity and fine granularity. From Fig.8a we can see that, even if the number of the attributes in an access policy reach 50, the size of the policy is just less than 1 kilobyte. This is because the attributes are actually some numbers or words to describe the characteristic of a certain device. Each digit or letter occupies only 1 byte in the standard character encoding of electronic communication. It is obvious that the storage overhead of access policy is reasonable and acceptable.
- **Session keys** In our access control scheme, each pair of communication participants need to first authenticate each other and then agree on a session key used for the subsequent communication. For convenience, they may specify an expire time and reserve the session key for some time. The session key can be generated by any standard identity-based authentication and key agreement protocol. For example, we use identity-based cryptographic algorithm to generate a 128 bits session key and use AES-128 to ensure the subsequent communication. The storage overhead of session key is linear to the number of communication participants and almost negligible in our scheme, as shown in Fig.8b.

B. COMPUTATION OVERHEAD

Undoubtedly, Hyperledger Fabric is one of the most promising platforms for distributed ledger solutions. Whether academia or industry has already provided extensive analysis on the performance of this well-known platform. Hence, to avoid repetition, our analysis on the computation overhead mainly focuses on the new component introduced in our scheme, the access control protocol part. We will analyze how it will affect the overall performance below. The analysis results are as shown in Fig.9.

From Fig.4 we can see that, using each proper secret key to sign the random number selected by Bob contributes to the major computation overhead of Alice. Each such secret key is corresponding to an attribute matching the access policy of Bob. Hence, as the number of attributes increases, so does the computation overhead and it is linear to the number of attributes. For Bob, in addition to verifying the signatures provided by Alice, he needs to hash each corresponding PK along with the ID of Alice to get each address which has been issued the satisfied attribute. It is obvious that the computation overhead of Bob is also linear to the number of attributes which are owned by Alice and match the access policy at the same time. A quality C++ implementation of elliptic curve digital signature algorithm (ECDSA) typically spends 2.87ms

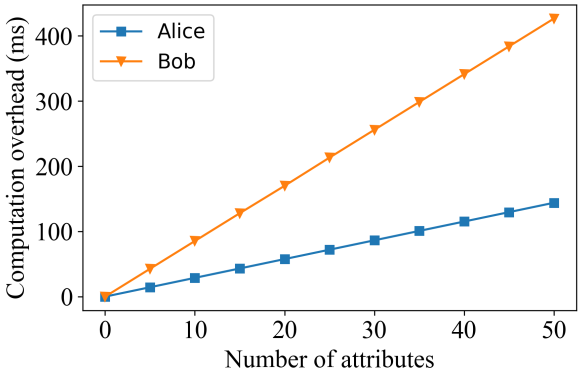


FIGURE 9. **Computation overhead of Alice and Bob.**

to compute a NIST256P signature and 6.34ms to verify it. The confidentiality and authenticity of the communication between Alice and Bob are ensured by the session key using AES-128 algorithm. From Fig.9 we can see that, although the computation overhead of Bob is higher than that of Alice as the increase in the number of attributes, the computation overhead for such IoT devices is reasonable and acceptable. That is to say, our attribute-based access control scheme using blockchain could be efficiently applied in IoT scenarios.

VII. CONCLUSION

In this paper, we proposed a novel attribute-based access control scheme using blockchain technology to improve the access management for billions of resource-constrained devices in IoT. This decentralized and scalable access control system solved the problem of the lack of trust and made the system more robust. We defined a new type of transactions which record the authorization of attributes. The IoT devices in our design are independent to the consensus process of blockchain network which significantly decreases the overall computation and communication overhead. What's more, some parts of the proposed scheme, such as the consensus algorithm, the AKA protocol, are modular design, which greatly enhances the flexibility of the system and facilitates the future maintenance and update. The security analysis proved our scheme to be secure in practical application and the simulation experiments demonstrated that it is effective and efficient to enforce strict and fine-grained access control in IoT.

REFERENCES

[1] M. Iansiti and K. R. Lakhani, "The truth about blockchain," *Harvard Bus. Rev.*, vol. 95, no. 1, pp. 118–127, 2017.

[2] O. Salman, S. Abdallah, I. H. Elhajj, A. Chehab, and A. Kayssi, "Identity-based authentication scheme for the Internet of Things," in *Proc. Comput. Commun.*, Jun. 2016, pp. 1109–1111.

[3] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "PAuthKey: A pervasive authentication protocol and key establishment scheme for wireless sensor networks in distributed IoT applications," *Int. J. Distrib. Sensor Netw.*, vol. 2014, no. 2, pp. 561–563, 2014.

[4] V. Shivraj, M. Rajan, M. Singh, and P. Balamuralidhar, "One time password authentication scheme based on elliptic curves for Internet of Things (IoT)," in *Proc. IEEE 5th Nat. Symp. Inf. Technol., Towards New Smart World (NSITNSW)*, Feb. 2015, pp. 1–6.

[5] J. L. Hernández-Ramos, A. J. Jara, L. Marín, and A. F. S. Gómez, "DCapBAC: Embedding authorization logic into smart things through ECC optimizations," *Int. J. Comput. Math.*, vol. 93, no. 2, pp. 345–366, 2016.

[6] D. Hardt, *The OAuth 2.0 Authorization Framework*, document RFC 6749, Oct. 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6749.txt

[7] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, and G. Ferrari, "IoT-OAS: An Oauth-based authorization service architecture for secure services in IoT scenarios," *IEEE Sensors J.*, vol. 15, no. 2, pp. 1224–1234, Feb. 2015.

[8] D. Ferraiolo, D. R. Kuhn, and R. Chandramouli, *Role-Based Access Control*. Norwood, MA, USA: Artech House, 2003.

[9] V. C. Hu *et al.*, "Guide to attribute based access control (ABAC) definition and considerations (Draft)," NIST, Gaithersburg, MD, USA, Tech. Rep. NIST SP 800-162, 2013.

[10] N. Ye, Y. Zhu, R.-C. Wang, R. Malekian, and L. Qiao-Min, "An efficient authentication and access control scheme for perception layer of Internet of Things," *Appl. Math. Inf. Sci.*, vol. 8, no. 4, pp. 1617–1624, 2014.

[11] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.

[12] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Inf. Sci.*, vol. 379, pp. 42–61, Feb. 2017.

[13] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Tech. Rep., 2008.

[14] A. Ouaddah, H. Mousannif, A. A. Elkalam, and A. A. Ouahman, "Access control in the Internet of Things: Big challenges and new opportunities," *Comput. Netw.*, vol. 112, pp. 237–262, Jan. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128616303735

[15] T. Hardjono and A. S. Pentland, "Verifiable anonymous identities and access control in permissioned blockchains," Tech. Rep., 2016.

[16] M. Conoscenti, A. Vetrò, and J. C. D. Martin, "Blockchain for the Internet of Things: A systematic literature review," in *Proc. IEEE/ACS 13th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2016, pp. 1–6.

[17] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "FairAccess: A new Blockchain-based access control framework for the Internet of Things," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5943–5964, 2017.

[18] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, Apr. 2018.

[19] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. OSDI*, vol. 99. 1999, pp. 173–186.

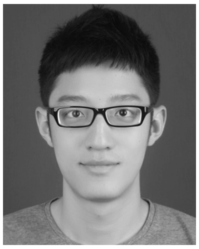
[20] D. Ongaro and J. K. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf.*, 2014, pp. 305–319.

[21] D. Basin, S. Mödersheim, and L. Vigano, "OFMC: A symbolic model checker for security protocols," *Int. J. Inf. Secur.*, vol. 4, no. 3, pp. 181–208, 2005.

[22] M. Turuani, "The CL-Atse protocol analyser," in *Proc. Int. Conf. Rewriting Techn. Appl.*, 2006, pp. 277–286.

[23] A. Armando and L. Compagna, "SATMC: A SAT-based model checker for security protocols," in *Proc. Eur. Workshop Logics Artif. Intell.*, 2004, pp. 730–733.

[24] Y. Boichut, P.-C. Héam, O. Kouchnarenko, and F. Oehl, "Improvements on the Genet and Klay technique to automatically verify security protocols," in *Proc. AVIS*, vol. 4, 2004, pp. 1–84.



SHENG DING received the B.Eng. degree in information security from Xidian University, China, in 2012, where he is currently pursuing the Ph.D. degree with the School of Cyber Engineering. His current research interests include cryptography, data security, and access control.

JIN CAO received the B.Sc. and Ph.D. degrees from Xidian University, China, in 2008 and 2015, respectively, where is currently an Associate Professor. His research interests include wireless network security and application cryptography.

CHEN LI received the Ph.D. degree in cryptography from Xidian University, China, in 2015, where he is currently a Postdoctoral Fellow of the School of Telecommunications Engineering. His current research interest is cryptography.

KAI FAN received the B.S., M.S., and Ph.D. degrees from Xidian University, China, in 2002, 2005, and 2007, respectively, where he is currently an Associate professor with the State Key Laboratory of Integrated Service Networks. He has published over 40 papers in journals and conferences. He held three Chinese patents. He has managed five national research projects. His research interests include cloud computing security, the IoT security, and information security.

HUI LI received the B.S. degree from Fudan University, in 1990, and the M.S. and Ph.D. degrees from Xidian University, in 1993 and 1998, respectively, where he is currently a Professor with the School of Cyber Engineering. He was with the Department of Electrical and Computer Engineering, University of Waterloo, as a Visiting Scholar, in 2009. His research interests include the areas of cryptography, the security of cloud computing, wireless network security, and information theory. He has served as the TPC Co-Chair of ISPEC 2009 and IAS 2009, the General Co-Chair of E-Forensic 2010, ProvSec 2011, and ISC 2011, and the honorary Chair of NSS 2014 and ASIACCS 2016.