# Edge Blockchain Assisted Lightweight Privacy-preserving Data Aggregation for Smart Grid

Weifeng Lu, Zhihao Ren, Jia Xu, *Member, IEEE,* Siguang Chen, *Member, IEEE*

***Abstract*—Compared with traditional power systems, smart grid is designed to provide effective and secure energy services. Data aggregation is one of the key technologies in wireless sensor networks, which reduces the amount of data transmission between nodes by merging similar data and simplifying redundant data, thus significantly reducing the computation cost and communication overhead of the system. Many data aggregation schemes have been developed for the smart grid in the past years. However, most of the data aggregation schemes ignore the data security and privacy protection issues of the edge layer. To solve these problems, in this paper, we propose an edge blockchain assisted lightweight privacy-preserving data aggregation for smart grid, named EBDA. In this work, we integrate edge computing and blockchain to design a three-layer architecture data aggregation scheme for smart grid. This new architecture supports a two-level data aggregation scheme, which is more efficient and secure. Through theoretical analysis and simulations, EBDA shows great superiority in terms of resisting network attacks, reducing system computation costs and communication overhead compared with existing schemes.**

***Index Terms*—edge computing; blockchain; smart grid; data aggregation.**

## I. Introduction

As the deep integration of the traditional power industry and modern information technology, smart grid has attracted extensive attention from both academia and industry. As the next-generation power network [1], [2], smart grid can provide efficient and intelligent power distribution as well as the information exchange between users and control center using advanced information technology and wireless communication technology [3], [4]. In the smart grid, users can interact with power companies via smart meters. The smart meter monitors the customer's home power consumption in real-time and transmits the data to the control center, which adjusts the power distribution and price accordingly. The users can also change their power consumption habits in time according to smart meters. However, due to the large-scale deployment of smart meters, a large amount of monitoring data will be generated and transmitted, which will bring great computation cost and communication overhead to the smart grid.

This work has been supported in part by the NSFC (No. 61872193, 61971235), the China Postdoctoral Science Foundation (No. 2018M630590) and NUPTSF (Grant No. NY220056).

W. Lu, Z. Ren and J. Xu are with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, China. (email: luwf@njupt.edu.cn; 153532300@qq.com; xujia@njupt.edu.cn).

S. Chen is with the College of IoT, Nanjing University of Posts and Telecommunications, Nanjing, China. (email: siguang1984@126.com).

Corresponding author: Jia Xu. (e-mail: xujia@njupt.edu.cn).

Edge computing [5] is a promising distributed computing paradigm, which provides applications with converged computing, storage, and network resources at the edge of the network close to the source of things or data. At the same time, edge computing is also an enabling technology. By providing these resources at the edge of the network, it can meet the key needs of the industry in agile connection, real-time business, data optimization, application intelligence, security and privacy-preserving. Some previous work [6]–[9] have shown that edge computing models have advantages over traditional cloud computing models in real-time data processing and analysis, high security, privacy-preserving, strong scalability, location awareness and low traffic. Therefore, the edge computing has received extensive attention and unanimous recognition from industry and academia. Taking edge computing based smart grid as an example, edge server can pre-process the collected smart meter data and then upload the pre-processed data to the cloud server. The cloud server can effectively adjust the power price and distribution through the control strategy. However, due to the limited resources of network edge devices, the existing data security protection methods are not fully applicable to edge computing architecture for edge devices with limited resources. Moreover, the highly dynamic environment at the edge of the network will also make the network more vulnerable and difficult to protect. Therefore, Stojmenovic *et al.* [10] and Roman *et al.* [11] have pointed out that it is a very important research direction to realize security, high-performance collaboration and privacy-preserving among edge servers.

In this paper, we consider the smart grid scenario based on the edge computing. The data monitored by the smart meter is uploaded to the local edge server, which performs the local aggregation. Then, the specific edge server performs the global aggregation and adds the information to the blockchain. The control center can obtain the global aggregated plaintext by reading the information in the blockchain. Therefore, the control center can use smart meters to obtain the power consumption of local area, so as to adjust the power price in real time. In this case, we should guarantee that the global result can only be obtained by the control center, and the privacy of the user should be ensured. To achieve this goal, we propose the Edge Blockchain assisted lightweight privacy-preserving Data Aggregation for smart grid (EBDA). The main contributions of this paper are summarized as follows:

- We propose EBDA, which combines the homomorphic Paillier encryption and one-way hash chain techniques so that the edge servers can not only save communication overheads by aggregating data from the same region, but also can filter false data in advance, improving the secure performance and robustness.

- We introduce the blockchain into the edge layer to improve the security performance of edge layer. We present a detailed analysis to show that EBDA can improve the ability to resist attacks adaptably for various deployment environments.
- We theoretically analyze the computation cost and communication overhead of EBDA and conduct simulations to verify the performance of the proposed scheme. Our simulations show that EBDA achieves the significant reduction in both computation cost and communication overhead compared with the comparison schemes.

The rest of this paper is organized as follows: Section II describes the related work. Section III describes the system model and the design goal. Section IV describes some preliminaries. Section V describes the EBDA in detail. Section VI describes theoretical analysis of the EBDA. Section VII describes performance evaluation of the EBDA. Finally, we conclude this paper in Section VIII.

## II. Related Work

Data aggregation technology means that the intermediate nodes in the network do not directly transmit data after receiving the data from the predecessor node, but preprocess the received data, and forward the calculated single-dimensional data to the successor node. Since data aggregation aggregates multidimensional data into single-dimensional data according to certain operations, data aggregation can reduce data redundancy and communication overhead. Since the computing power and bandwidth resources of nodes are limited, data aggregation can increase the lifecycle of the network. Data aggregation is one of the key technologies in the Internet of Things. In addition, the privacy protection nature of data aggregation can ensure the privacy of sensitive data in the process of data aggregation, which has become a research hotspot. In this section, we describe some data aggregation schemes for privacy-preserving that are similar to our EBDA application scenarios.

To protect the data integrity of wireless sensor networks, Shen *et al.* [12] proposed an identity-based converged signature algorithm that both preserved data integrity and reduced the communication overhead and storage costs of wireless sensor networks. Wang *et al.* [13] proposed an identity-based smart grid data aggregation protocol that prevented not only unauthorized reads and fine-grained analysis, but also unexpected false and maliciously modified messages. From the perspective of resource limitation in edge computing scenarios, Zhang *et al.* [14] proposed an efficient privacy protection data aggregation scheme, which transformed time-consuming signature operations into offline processing, thus significantly reducing the load of online computing. Although the above schemes can well reduce the system's latency and communication overhead, they still have a great security risk to some extent. When the edge server receives the data from the user layer, the external attacker may directly attack the edge server and try to obtain the user's private data from the edge servers. Therefore, servers deployed at the edge of the network cannot be fully trusted.

TABLE I
Features of EBDA and existing works

| Features \ Ref. | EBDA | [13] | [14] | [17] | [19] | [26] |
|---|---|---|---|---|---|---|
| Computation cost | Y | Y | Y | Y | N | Y |
| Communication overhead | Y | N | Y | N | N | N |
| Security | Y | Y | Y | Y | Y | Y |
| Robustness | Y | N | N | N | N | N |

The emergence of blockchain [15], [16] technology brings new ideas to solve the trust crisis, the blockchain is decentralized and tamper-proof. Guan *et al.* [17] proposed a smart grid user privacy protection and data aggregation scheme based on blockchain. To solve the problem of transaction security in distributed smart grid energy transactions that did not rely on trusted third parties, Aitzhan and Svetinovic [18] proposed a scheme to use blockchain, anonymous encrypted message flow and multiple signatures to realize the proof of concept of a decentralized energy trading system, enabling the peers to execute transactions securely. Liang *et al.* [19] proposed a new distributed protection framework based on blockchain to improve the self-defense capability of modern power system against network attacks. They also discussed comprehensively how to use blockchain to enhance the robustness and security of the smart grid.

There are several recent works that address the integration of edge computing and blockchain to improve the performance of data processing, task offloading, and distributed control system, etc. [20]–[22]. Nguyen *et al.* [23] proposed a blockchain network to transfer data processing and mining tasks to mobile edge servers through wireless channels. Casado-Vara *et al.* [24] proposed a new architecture with edge computing layer and an algorithm based on blockchain to improve data quality and false data detection. Stanciu [25] studied the application of blockchain technology as an edge computing platform to realize distributed control system. In the field of data aggregation, Wang *et al.* [26] recently proposed a blockchain based secure data aggregation strategy to restrict task receivers to search and accept tasks. This work integrated blockchain and edge computing to obtain high throughput and low transaction latency of data aggregation task allocation. However, the performance of data aggregation is not improved through the integration of blockchain and edge computing.

In this paper, we integrate blockchain and edge computing techniques in data aggregation scheme. By integrating edge computing, EBDA can take full advantage of local computing resources, reducing the computation cost and communication overhead. We also introduce the blockchain into the edge layer to improve the security performance of edge devices and the ability to resist attacks adaptably for various deployment environments. We compare the features of our EBDA with the existing works in Table I.

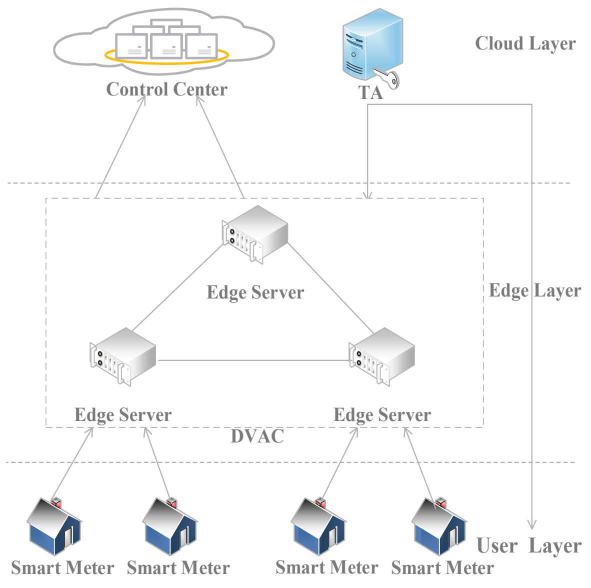


Fig. 1. EBDA network model

## III. System Model and Design Goal

In this section, we present the system model, attack model, and our design goals.

### A. System Model

We consider an edge blockchain assisted smart grid system, which is composed of three layers: the user layer, the edge layer, and the cloud, as shown in Fig. 1:

*1) The cloud layer:* The cloud is composed of the control center (CC) and the trusted third authority (TA). TA is a fully trusted third authority, so it plays a critical role in the whole system. TA distributes public parameters and keys to all entities in the power system. After completing the specified tasks, TA will go offline. CC has the ability to read, analyze, store and manage the aggregated ciphertext provided by the edge blockchain automatically in real-time. Note that, to facilitate the control center to monitor the power consumption at a certain time, the control center reads the data every $\eta$ minutes.

*2) The edge layer:* The edge layer divides the smart grid coverage into $n$ regions, that is, the data of each region will be pre-processed by the specific edge server. In order to guarantee the security and integrity of data in the edge layer, our scheme adopts the Distributed Voting Authorization Certificate (DVAC) in the edge layer. During the system initialization phase, each edge server broadcasts its computing resources and network status to other edge servers. When each edge server receives these information, it votes on the edge servers based on these information. It is worth noting that each edge server has only $w$ voting rights and cannot vote on the same edge server more than once. TA is responsible for counting the voting results, selecting $w$ edge servers with the highest voting results as candidate nodes, and broadcasting the results to these selected nodes. These selected candidate nodes will negotiate a block right ownership order based on their network status. We define the master node as a candidate node with the block right ownership. When the candidate node is selected as the master node, it first collects the transaction information from the whole edge layer, then packages the transaction information into the block, and finally broadcasts the block to the other candidate nodes. The other candidate nodes validate the block and vote on it. If most of the candidate nodes agree on its validity, the block becomes an irreversible block in the blockchain.

*3) The user layer:* The user layer is a further division of the $n$ regions of the smart grid. We consider that there are $m_i$ households in region $i$, and each household deploys a smart meter to monitor its power consumption. We denote $ES_i$ as the edge server in $i-th$ region, and $SM_{ij}$ as the $j-th$ smart meter in region $i$, where $j <= m_i$. $SM_{ij}$ will measure the user's power consumption data $d_{ij}$ and then transmit the data to $ES_i$.

### B. Attack Model

In our attack model, we assume that there are some external attackers. These attackers try to break into smart grid systems and try to gain access to users' private data. We refer to the attack as forgery data injection attack, which can be launched in two ways. First, the attackers may eavesdrop on or tamper with user information in the communication channels between nodes. Second, the attackers may break down nodes and directly access or modify the database information in nodes. Next, we discuss the possible attacks in the user layer and the edge layer, respectively.

*1) The user layer:*

In the user layer, attackers may launch the first kind of forgery data injection attack on the communication channel between smart meters and edge servers. The attackers can eavesdrop or modify the information in the channels.

In addition, attackers try to launch the second kind of forgery data injection attack to directly attack the smart meter and want to access or modify the users private data.

*2) The edge layer:*

In the edge layer, attackers may launch the first kind of forgery data injection attack on the communication channel between edge servers. First, when the edge server transmits local aggregated data to the master node, attackers may break into the communication channel and tamper with the local aggregated data results. Second, when the master node generates and broadcasts the new block to the candidate nodes. At this time, the attackers may invade the broadcast channel and modify the block content. Moreover, the candidate node verifies the block and sends the verification to the master node. At this time, attackers may invade the communication channel between two nodes and attempt to tamper with the voting result.

In addition, attackers try to launch the second kind of forgery data injection attack to directly attack the edge server and want to access or modify the user's private data.

### C. Design Goal

Our design goal is to design a data aggregation for smart grid satisfying the following desirable goals:

- Confidentiality: Confidentiality is a fundamental requirement in data transmission. Even if an attacker can eavesdrop on communication channel, which cannot achieve access to the user's private information.
- Integrity and Authentication: Integrity is used to ensure that data is not tampered with during transmission. Authentication ensures the validity of the data source, that is, encrypted reports are generated by legitimate users. When the report is read, both the edge server and the control center can detect any unauthorized or modified report.
- Robustness: The proposed scheme should be robust, so that even if some smart meters fail at some point, the edge server can still complete the data aggregation, and further provide the power consumption in the corresponding area for the control center.
- Privacy-preserving: Attackers cannot directly access the smart meter's user privacy data, even if they successfully launch forgery data injection attack. Although the control center can obtain the aggregated plaintext in the blockchain, it cannot directly obtain the user's data.
- Efficiency: The resource utilization of the proposed scheme should be efficient, that is, the computation cost and communication overhead of the power system should be as small as possible.

Frequently used notations in this paper is given in Table II.

TABLE II
FREQUENTLY USED NOTATION

| Notations | Definition |
|---|---|
| $ES_i$ | Edge server in region $i$ |
| $SM_{ij}$ | $j$-th smart meter in region $i$ |
| $k$ | Length of bilinear pairings parameters |
| $k^+$ | Length of prime numbers |
| $l$ | Length of one-way hash chain parameters |
| $h$ | Hash function of one-way hash chain |
| $H_1, H_2$ | Hash functions in data aggregation and signature |
| $G_1, G_2$ | Cyclic addition group, Cyclic multiplication group |
| $(N, g)$ | Public key of Paillier encryption algorithm |
| $(\lambda, \mu)$ | Private key of Paillier encryption algorithm |
| $\overrightarrow{a} = (a_1, a_2, ..., a_n)$ | Super-linear sequence |
| $m_i$ | Number of smart meters in each region $i$ |
| $HC_j^i$ | One-way hash chain of $SM_{ij}$ |
| $h_{ij\_s}$ | Element of the hash chain of $SM_{ij}$ in time slot $s$ |
| $\phi_{ij}$ | Constraint parameter to improve the robustness of ciphertext of $SM_{ij}$ |
| $\partial_i$ | Computation resource status of $ES_i$ |
| $(N, g_i)$ | Public key in region $i$ |
| $c_{ij}$ | Ciphertext of user data of $SM_{ij}$ |
| $mac_{ijs}$ | Mask of ciphertext $c_{ij}$ in time slot $s$ |
| $C_i$ | Aggregation result of user data in region $i$ |
| $C$ | Aggregate result for all user data |
| $Seq_1$ | List of candidate nodes |
| $Seq_2$ | List of master nodes |

## IV. PRELIMINARIES

In this section, we briefly review the basic technologies used in our proposed scheme.

### A. Blockchain

Blockchain can be considered as a distributed storage ledger that records transactions in sequence with timestamps, which is maintained by many distributed nodes through consensus protocol. Blockchain has the following three important characteristics:

- Decentralization: Different from the existing centralized system, all nodes in the blockchain are peers, jointly responsible for maintaining the entire network and building a decentralized environment. The decentralized nature of blockchain can greatly improve the security performance of the whole system.
- Transparency and traceability: All transactions in the blockchain are public and distributed, and each node stores all transactions in the blockchain. Users can query transaction information through the public interface provided by blockchain. The blocks in the blockchain are linked in sequence through a linked list structure, ensuring that users can trace transactions along with the blockchain.
- Tamper-proof: Since the blocks are linked by a linked list structure, the block header of each block contains the hash address of the previous block. In addition, the hash address of each block is based on the block content. Therefore, if an attacker tries to modify the contents of the block, the attacker will pay a huge cost.

### B. One-Way Hash Chain

The one-way hash chain is a fundamental encryption technology, and is widely used in data flow verification, its structure is shown in Fig. 2.

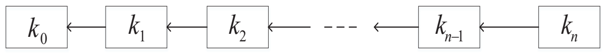


Fig. 2. The structure of the one-way hash chain

For given a secure hash function $h : \{0,1\}^* \rightarrow \{0,1\}^l$, a one-way hash chain is composed of a set of values $\{k_0, k_1, k_2, ..., k_n\}$ for $n \in Z$ where $k_n \in \{0,1\}^l$ is a randomly chosen value, and

$$k_i = h(k_{i+1}), i = 0, 1, 2, ..., n-1 \tag{1}$$

$l$ is responsible for ensuring the length of the one-way hash chain parameters. One-way hash chain has the following characteristics, for given $k_i$ in a hash chain, it can compute $k_j$, where $j < i$; however, it is almost impossible to compute $k_l$, where $l > i$.

### C. Bilinear Pairings

Let $G_1, G_2$ be a cyclic addition group and a cyclic multiplication group of prime order $q$, respectively. Let $P_0 \in G_1$ be a generator. We define $e$ as a bilinear pairing if $e : G_1 \times G_1 \rightarrow G_2$ is a mapping satisfying the following properties:

- Bilinear: For all $a, b \in Z_q^*$,

$$e(aP_0, bP_0) = e(P_0, P_0)^{ab} \tag{2}$$

- Non-degenerate: $P_0$ should satisfy $e(P_0, P_0) \neq 1_{G_1}$
- Computable: For all $P_0, Q \in G_1$, $e(P_0, Q)$ should be computable.

### D. Paillier Encryption Algorithm

Paillier encryption has the homomorphic encryption property, which is widely required in many privacy protection applications. Note that Paillier encryption algorithm has provable security against the chosen plaintext attack, and its correctness and security have been proved in [27]. Specifically, the Paillier encryption algorithm is composed of three parts, namely: key generation, encryption, and decryption.

- Key Generation: Given a security parameter $k^+$, choose two large primes $p_1, q_1$, where $|p_1| = |q_1| = k^+$, then compute

$$N = p_1 q_1 \tag{3}$$

and

$$\lambda = lcm(p_1 - 1, q_1 - 1) \tag{4}$$

, define a function

$$L(u) = \frac{u - 1}{N} \tag{5}$$

, choose a generator $g \in Z_{N^2}^*$,

$$\mu = (L(g^\lambda \bmod N^2))^{-1} \bmod N \tag{6}$$

is further calculated. Finally, the public key $pk = (N, g)$, and the corresponding private key $sk = (\lambda, \mu)$ are obtained.

- Encryption: For a given message $M \in Z_N$, choose a random number $r \in Z_N^*$, and calculate the ciphertext

$$c = E(M) = g^M \cdot r^N \bmod N^2 \tag{7}$$

- Decryption: For the given ciphertext $c \in Z_{N^2}^*$, the corresponding plaintext is

$$M = D(c) = L(c^{\lambda \bmod N^2}) \cdot \mu \bmod N \tag{8}$$

## V. EBDA Scheme

In this section, we propose EBDA for modern smart grid systems. This section is specifically divided into system initialization, user data report generation, edge layer data processing, and control center data analysis. The system initialization phase is primarily responsible for assigning parameters to other entities in the system. User data report generation, edge layer data processing and control center data analysis correspond to the three-layer architecture given in our system model. The data information in the user data report generation stage will be transmitted to the edge server, and the data will be added to the blockchain after the edge server completes the data pre-process. The control center will get the content it wants by reading the blockchain, and then process the data through its private key. The process of EBDA is shown in Fig.3.

### A. System Initialization

*1) Paillier parameter generation:* Let $k$, $k^+$ be security parameters. TA generates bilinear pairings $(q, P_0, G_1, G_2, e)$ by running $gen(k)$, where $k$ is length of bilinear pairings parameters. Then TA calculates the Paillier algorithm public key $(N = p_1 q_1, g)$, and the corresponding private key $(\lambda, \mu)$, where $p_1, q_1$ are two large primes with $|p_1| = |q_1| = k^+$. $k^+$ is the length of the large prime numbers.

*2) Operation parameter generation:* The length of the data $d_{ij}$ measured by each smart meter will not be greater than the fixed value $d$. Then, TA chooses a super-linear sequence $\overrightarrow{a} = (a_1, a_2, ..., a_n)$, where $a_1 = 1$ and $a_2, ..., a_n$ are large primes. Finally, TA selects a set of sequences $(g_1, g_2, ..., g_n)$, where

$$g_i = g^{a_i}, \ i = 1, 2, ..., n \tag{9}$$

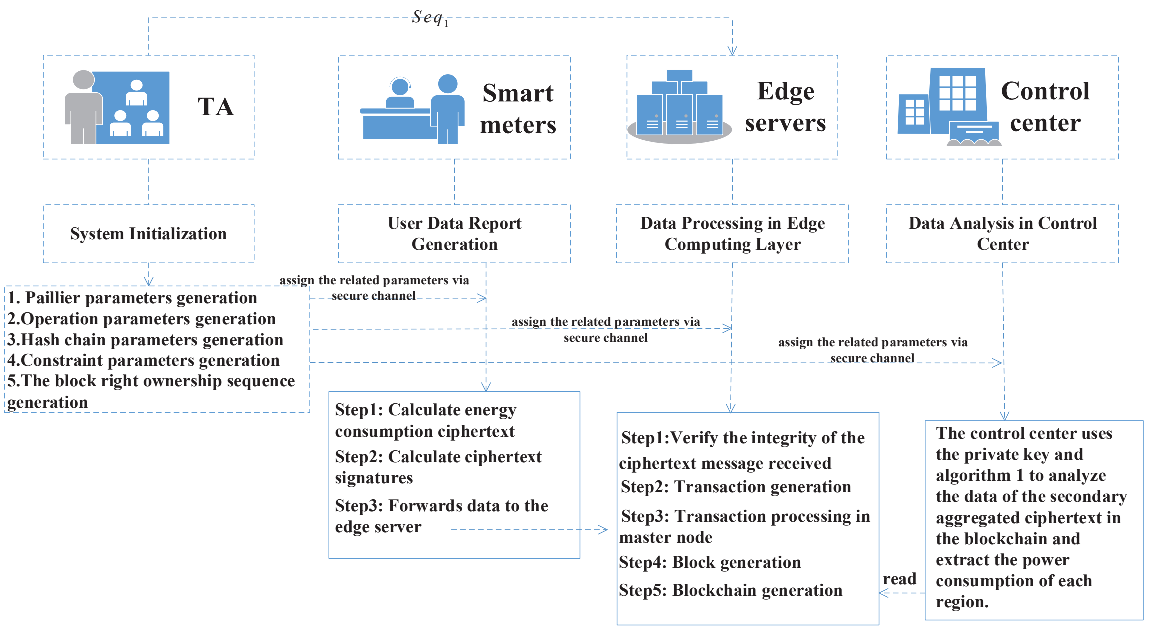
$(N, g_i)$ is the public key corresponding to each $ES_i$ region, and the value of each $g_i$ is determined by $a_i$.

To ensure the security of edge server identities, TA selects a random prime number $x_i$ used as the private key for each edge server $ES_i$, and calculates its public key $y_i = x_i P_0$. In addition, TA chooses two secure cryptographic hash functions $H_1, H_2$, where $H_1 : \{0,1\}^* \rightarrow Z_N^*, H_2 : \{0,1\}^* \rightarrow G$.

*3) Hash chain parameter generation:* To ensure the security of the smart meter $SM_{ij}$, TA selects a security hash function $h$, where $h : \{0,1\}^* \rightarrow \{0,1\}^l$. In the proposed scheme, the control center reads the entire area's power data every $\eta$ minutes. To ensure data continuity, $SM_{ij}$ needs to generate the user's power consumption with a smaller time interval. Here, we further divide $\eta$ into $w$ time slots, as shown in Fig. 4.

At each time slot, each $SM_{ij}$ generates power consumption of users. Therefore, we must ensure that data are secure and can be correctly received by edge servers. To satisfy this requirement, TA builds $\sum_{i=1}^{n} m_i$ one-way hash chains $HC_1^i, HC_2^i, ..., HC_{m_i}^i$, where $i = 1, 2, ..., n$. The length of each chain

$$HC_j^i = \{h_{ij\_0}, h_{ij\_1}, ..., h_{ij\_w}\} \tag{10}$$



$Seq_1$
TA
Smart meters
Edge servers
Control center
System Initialization
User Data Report Generation
Data Processing in Edge Computing Layer
Data Analysis in Control Center
1. Paillier parameters generation
2.Operation parameters generation
3.Hash chain parameters generation
4.Constraint parameters generation
5.The block right ownership sequence generation
assign the related parameters via secure channel
assign the related parameters via secure channel
assign the related parameters via secure channel
Step1: Calculate energy consumption ciphertext
Step2: Calculate ciphertext signatures
Step3: Forwards data to the edge server
Step1:Verify the integrity of the ciphertext message received
Step2: Transaction generation
Step3: Transaction processing in master node
Step4: Block generation
Step5: Blockchain generation
The control center uses the private key and algorithm 1 to analyze the data of the secondary aggregated ciphertext in the blockchain and extract the power consumption of each region.
read

Fig. 3. The workflow of EBDA

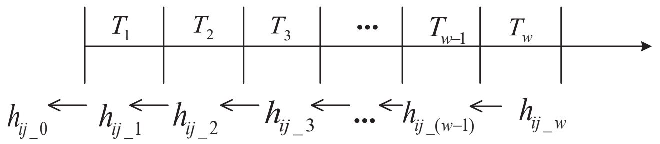


Fig. 4. Key generation and hash chain with time slots

is $w+1$, $j \in \{1,2,...,m_i\}$, where $h_{ij\_w} \in \{0,1\}^l$ is a randomly selected number, and

$$h_{ij\_s} = h(h_{ij\_(s+1)}||T_{s+1}) \quad s = 0,1,2,...,w-1 \tag{11}$$

$h_{ij\_s}$ is used for data authentication in time slot $T_s$. TA will digitally sign the $\sigma_{i0}$ on the heads $(h_{i1\_0}, h_{i2\_0}, ..., h_{im_i\_0})$ of all hash chains in the same area to ensure the integrity of these hash chains.

*4) Constraint parameter generation:* TA runs the pseudo-random generator and generates a random numbers $\phi_{ij} \in Z_N$, where $i = 1,2,...,n$, $j = 1,2,...,m_i$, as a constraint parameter for each $SM_{ij}$ and computes

$$\phi_{i0} = -(\phi_{i1} + \phi_{i2} + \cdots + \phi_{im_i}) \bmod N \tag{12}$$

as a constraint parameters of $ES_i$. Note that, $\phi_{ij}$ and $\phi_{i0}$ satisfy $\sum_{j=0}^{m_i} \phi_{ij} \equiv 0 \bmod N$.

*5) The block right ownership sequence generation:* The edge server $ES_i$ uses $\partial_i$ to represent its computation resource status. Each edge server broadcasts its own $\partial_i$ in the edge layer. When the edge server has collected the status of computing resources from all other servers in the edge layer, the edge server votes on the network to select the $w$ edge servers as candidates nodes that it considers to have the best performance. TA is responsible for statistical analysis of the voting results among the servers, and finally determines the $w$ edge servers with the best performance. Then, TA passes this list $Seq_1$ to these edge servers that have won the election. These $w$ edge servers negotiate the order of block right ownership $Seq_2$ at different moments according to their respective computing resource status, and broadcast the order to the edge layer.

Finally, TA chooses some of the above parameters as public parameters for the whole system:

$$\begin{aligned} pubs = \{&q, P_0, G_1, G_2, e, N, \\ &g_1, g_2, ..., g_n, H_1, H_2, h\} \end{aligned} \tag{13}$$

After completing the setting of the above parameters, the TA assigns the remaining parameters to each entity in the system model, as follows:

- For each $SM_{ij}$, $i \in \{1,2,...,n\}$, $j \in \{1,2,...,m_i\}$, TA assigns the secret hash chain (10), and the corresponding $\phi_{ij}$ to $SM_{ij}$ via a secure channel.
- For each $ES_i$, TA assigns secret key $(g_i, x_i)$, the constraint parameter $\phi_{i0}$, digital signature result $\sigma_{i0}$, and $(\phi_{i1}, \phi_{i2}, ..., \phi_{im_i})$ to $ES_i$.
- For the CC, TA assigns the private key $(\lambda, \mu)$ and function $L(\mu)$ to the control center.

## B. User Data Report Generation

At each time slot $T_s$, $s = 1,2,3,...,w$, each $SM_{ij}$ follows these steps to generate user power consumption $d_{ij}$ $(d_{ij} < d)$:

- Step 1: $SM_{ij}$ uses the corresponding secret keys $(g_i, N, \phi_{ij})$ to compute $c_{ij}$:

$$c_{ij} = g_i^{d_{ij}} \cdot H_1(T_s)^{\phi_{ij}} \bmod N^2 \tag{14}$$

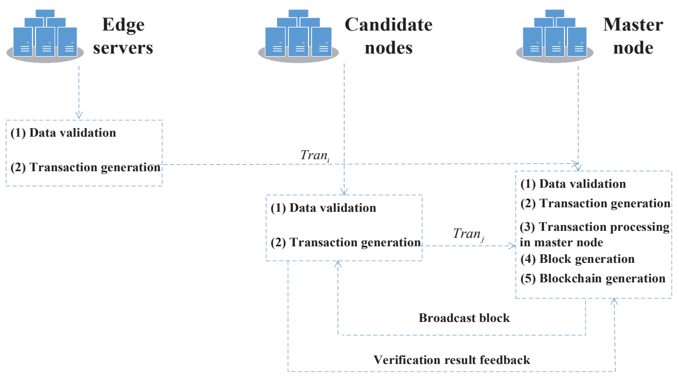

Edge servers
Candidate nodes
Master node
(1) Data validation
(2) Transaction generation
Tran_i
(1) Data validation
(2) Transaction generation
Tran_j
(1) Data validation
(2) Transaction generation
(3) Transaction processing in master node
(4) Block generation
(5) Blockchain generation
Broadcast block
Verification result feedback

Fig. 5. The execution process of data processing in edge layer.

- Step 2: When the data encryption is completed, $SM_{ij}$ uses $h_{ij\_s}$ in its hash chain $HC_j^i$ to compute $mac_{ijs}$, which can be considered a mask:

$$mac_{ijs} = h(c_{ij}||h_{ij\_s}) \tag{15}$$

- Step 3: $SM_{ij}$ forwards $(c_{ij}, h_{ij\_s}, mac_{ijs})$ to $ES_i$.

Note that, $SM_{ij}$ can run the above steps very efficiently specifically when $H_1(T_s)^{\phi_{ij}}$ is pre-computed in advance.

## C. Data Processing in Edge Layer

The servers of the edge layer mainly perform the data processing through the following five functions. Note that, all edge servers must perform the first two functions, but only the master node needs to perform the last three functions. The specific execution process is shown in Fig.5.

*1) Data validation:* Once receiving $(c_{ij}, h_{ij\_s}, mac_{ijs})$ in time slot $T_s$, each $ES_i$ verifies the validity of the data according to the following steps:

- Step1: As $ES_i$ has the authenticated $h_{ij\_0}$ from $\sigma_{i0}$, it can verify the validity of each $h_{ij\_s}$ in the hash chain $HC_j^i$.
- Step2: Once the edge server has successfully verified the validity of $h_{ij\_s}$, $ES_i$ verifies $c_{ij}$ by computing

$$mac_{ijs}{}' = h(c_{ij}||h_{ij\_s}) \tag{16}$$

and checking whether $mac_{ijs}{}' \overset{?}{=} mac_{ijs}$. If it does hold, the integrity of the ciphertext $c_{ij}$ is verified and received; otherwise, it will be filtered by $ES_i$.

*2) Transaction generation:* At each time slot $T_s$, once $ES_i$ finishes the data validation, each $ES_i$ aggregates the data set through the following formula:

$$\begin{aligned} C_i &= \prod_{j=1}^{m_i} c_{ij} \cdot H_1(T_s)^{\phi_{i0}} \bmod N^2 \\ &= \prod_{j=1}^{m_i} g_i^{d_{ij}} \cdot H_1(T_s)^{\phi_{ij}} \cdot H_1(T_s)^{\phi_{i0}} \bmod N^2 \\ &= g_i^{\sum_{j=1}^{m_i} d_{ij}} \cdot H_1(T_s)^{\sum_{j=0}^{m_i} \phi_{ij}} \bmod N^2 \end{aligned} \tag{17}$$

For the above equation, we can utilize the fact of $\sum_{j=0}^{m_i} \phi_{ij} \equiv 0 \bmod N$, to obtain $\sum_{j=0}^{m_i} \phi_{ij} = \beta N$. Then equation (17) can be represented as:

$$C_i = g_i^{\sum_{j=1}^{m_i} d_{ij}} \left(H_1(T_s)^{\beta}\right)^N \bmod N^2 \tag{18}$$

Now, $ES_i$ has completed the local data aggregation, and each edge server $ES_i$ digitally signs the aggregated data to ensure the integrity of the ciphertext:

$$\sigma_i = x_i H_2(C_i, ES_i, T_s) \tag{19}$$

Then, each $ES_i$ packages these results into a transaction in the form of

$$Tran_i = (C_i, ES_i, \sigma_i) \tag{20}$$

In our system model, $w$ edge servers are candidate nodes, which are responsible for maintaining and managing the edge blockchain. Since $w$ candidate nodes agree on the ownership sequence $Seq_2$ of blocks, different candidate nodes are responsible for generating blocks at each $T_s$ moment. The candidate node with block rights are also called the master node. Therefore, at time slot $T_s$, each $ES_i$ will transfer transaction $Tran_i$ to the master node.

*3) Transaction processing in master node:* Before producing the block, the master node needs to verify the received transactions through the following steps to ensure that all the transactions are valid.

At each time slot $T_s$, the master node collects the transaction information from all edge servers in the edge layer. To improve the validation capabilities, the master node randomly assembles all transactions into a new set

$$S = \{Tran_1, Tran_2, ..., Tran_n\} \tag{21}$$

$\lfloor n/2 \rfloor$ packages are randomly selected from $S$ to form the first subset $S_1$, and the remaining $\lceil n/2 \rceil$ packages form the second subset $S_2$. For master node, if the following equation is true, the transactions packages in $S_1$ are valid, otherwise the transactions are invalid.

$$e(P_0, \sum_{i=1}^{\lfloor n/2 \rfloor} \sigma_i) = \prod_{i=1}^{\lfloor n/2 \rfloor} e(y_i, H_2(C_i, ES_i, T_s)) \tag{22}$$

By using the verification method described above, the bilinear pairing operation costs for $S_1$ can be reduced from $2\lfloor n/2 \rfloor$ to $\lfloor n/2 \rfloor + 1$. Similarly, the master node verifies the transaction in $S_2$ through equation (23), and the bilinear pairing operation costs for $S_2$ can be reduced from $2\lceil n/2 \rceil$ to $\lceil n/2 \rceil + 1$.

$$e(P_0, \sum_{i=\lceil n/2 \rceil+1}^{n} \sigma_i) = \prod_{i=\lceil n/2 \rceil+1}^{n} e(y_i, H_2(C_i, ES_i, T_s)) \tag{23}$$

After transaction verification, the master node performs the global data aggregation operation through the following formula:

$$\begin{aligned} C &= \prod_{i=1}^{n} C_i \bmod N^2 \\ &= \prod_{i=1}^{n} g_i^{\sum_{j=1}^{m_i} d_{ij}} \left(H_1(T_s)^{\beta}\right)^N \bmod N^2 \\ &= g_1^{\sum_{j=1}^{m_1} d_{1j}} \cdot g_2^{\sum_{j=1}^{m_2} d_{2j}} \cdots g_n^{\sum_{j=1}^{m_n} d_{nj}} \left(\prod_{i=1}^{n} H_1(T_s)^{\beta}\right)^N \bmod N^2 \end{aligned} \tag{24}$$

As $(g_1, g_2, ..., g_n)$ satisfies the formula (9). So, the formula (24) can be simplified as:

$$\begin{aligned} &= g^{a_1 \sum_{j=1}^{m_1} d_{1j}} \cdot g^{a_2 \sum_{j=1}^{m_2} d_{2j}} \cdots g^{a_n \sum_{j=1}^{m_n} d_{nj}} \left(\prod_{i=1}^{n} H_1(T_s)^{\beta}\right)^N \bmod N^2 \\ &= g^{a_1 \sum_{j=1}^{m_1} d_{1j} + a_2 \sum_{j=1}^{m_2} d_{2j} + \cdots + a_n \sum_{j=1}^{m_n} d_{nj}} \left(\prod_{i=1}^{n} H_1(T_s)^{\beta}\right)^N \bmod N^2 \end{aligned} \tag{25}$$

*4) Block generation:* When the master node successfully obtains the global data aggregation ciphertext, it packages $(C, T_s)$ and adds it to the block header. In our scheme, the block consists of two parts: block header and block body. The block header contains the following attributes: block number ($block_{number}$), data aggregation result ($C$), Merkle Root ($Merkle\_root$), timestamp ($timestamps$), hash of previous block ($Hash_{previous}$), and hash of current block ($Hash_{current}$).

The block body records the transaction information of all edge servers at the time slot $T_s$. The transactions form a Merkle Tree, as shown in Fig. 6, where

$$Hash_i = H_2(Tran_i), i = 1, 2, ..., n \tag{26}$$

$$\begin{gathered} Hash_{xy} = H_2(Hash_x, Hash_y) \\ (x, y) = (1, 2), (3, 4), ..., (n-1, n) \end{gathered} \tag{27}$$

The Merkle Root is stored in the block header to ensure that the content of the block body is not tampered or forged.

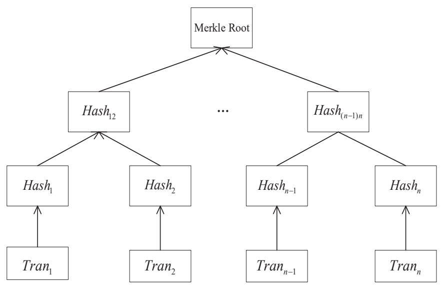


Fig. 6. Block body

When the transaction information is added to the block, the master node calculates the current hash address through the SHA-256 security encryption algorithm, as follows:

$$\begin{aligned} Hash_{current} = & SHA256(block_{number}, Hash_{previous}, C, \\ & timestamps, Merkle\_root) \end{aligned} \tag{28}$$

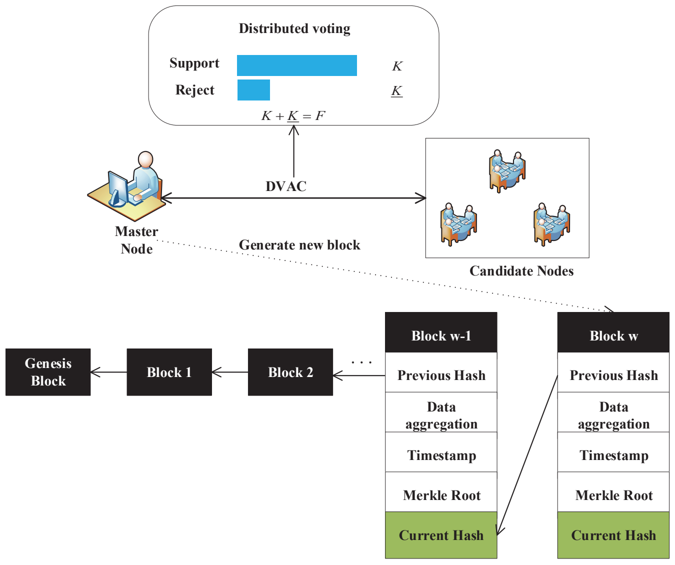


Fig. 7. The blockchain generation

*5) Blockchain generation:* To ensure the validity of the block, the block will be broadcast among all remaining candidate nodes. In this work, we use the DVAC consensus mechanism. The remaining candidate nodes judge the validity of the block, that is, whether the block content has been tampered with. Once most of the candidate nodes reach consensus, the block is considered valid and can be added to the blockchain.

For the consensus condition of the candidate nodes, we introduce the parameter $\tau$. Only when the voting result of the candidate node satisfies the following inequality, will the validity of the block be recognized.

$$\frac{K}{F} > \tau \tag{29}$$

The parameter $K$ represents the candidate nodes that recognize the validity of the block ($\underline{K}$ represents the number of remaining candidate nodes with opposite opinions), $F$ represents the number of candidate nodes, and $\tau$ is a threshold that must be greater than 50% to ensure that the majority of nodes in the blockchain network reach consensus. The blockchain generation process is shown in Fig. 7.

## D. Data Analysis in Control Center

The control center reads the blockchain information every $\eta$ minutes. The control center decrypts global aggregation ciphertexts in the block header through the Paillier decryption algorithm. To make it easy to decrypt the aggregation ciphertext, we define symbols $M$ and $R$:

$$M = a_1 \sum_{j=1}^{m_1} d_{1j} + a_2 \sum_{j=1}^{m_2} d_{2j} + \cdots + a_n \sum_{j=1}^{m_n} d_{nj} \tag{30}$$

$$R = \prod_{i=1}^{n} H_1(T_s)^{\beta} \tag{31}$$

Then the ciphertext $C$ can be converted into the form of:

$$C = g^M \cdot R^N \bmod N^2 \tag{32}$$

The final aggregation ciphertext still follows the format of Paillier encryption, so the control center can use the private keys $(\lambda, \mu)$ and $L(\mu)$ to execute the Paillier decryption to obtain the aggregation plaintext $M$:

$$M = D(C) = L(C^{\lambda \bmod N^2}) \cdot \mu \bmod N \tag{33}$$

The ultimate goal of control center is to obtain the fine-grained power consumption of each area. To achieve this goal, the control center uses Algorithm 1 to obtain area data, and extracts $(D_1, D_2, ..., D_n)$ from $M$, where

$$D_i = \sum\nolimits_{j=1}^{m_i} d_{ij}, i = 1, 2, ..., n \tag{34}$$

**Algorithm 1** Aggregated area report extraction

```
Input: $M$ and $\overrightarrow{a} = (a_1, a_2, a_3, ..., a_n)$;
Output: $(D_1, D_2, ..., D_n)$;
1: Let $X_n = M$;
2: for $i = n$; $i > 1$; $i - -$ do
3:     $X_{i-1} = X_i \bmod a_i$;
4:     $D_i = \frac{X_i - X_{i-1}}{a_i} = \sum_{j=1}^{m_i} d_{ij}$;
5:     $D_1 = X_1 = \sum_{j=1}^{m_1} d_{1j}$;
6: end for
7: return $(D_1, D_2, ..., D_n)$;
```

## VI. Theoretical analysis

In this section, we present the theoretical analysis, demonstrating that EBDA can achieve the design goals of confidentiality, privacy-preserving, authentication, integrity, and fault-tolerance.

### A. Confidentiality and Privacy-preserving

In the user data report generation phase, each $d_{ij}$ sensed by $SM_{ij}$ and the constraint parameter $\phi_{ij}$ are added in the Paillier encryption algorithm to get the ciphertext $c_{ij}$. Meanwhile, the local data aggregation operations utilize the additive homomorphic attributes to aggregate the ciphertext $c_{ij}$ in the same area, and the form of aggregated ciphertext can be expressed as $C_i$. Similarly, in the global data aggregation operation phase, we can get $C$. According to (30) and (31), we can find the final aggregated ciphertext (32), which follows the Paillier encryption system. Note that, the Paillier Cryptosystem is provably secure against chosen-plaintext attack [27]. Therefore, the confidentiality of both individual power consumption $d_{ij}$ and aggregated plaintext $M$ are guaranteed.

Even if some attackers could break into the communication channel between the user layer and the edge layer or between the edge layer servers, these attackers cannot get any private data about the user from the encrypted ciphertext. On the one hand, in our system model, although the edge server collects the data set from all the smart meters in the same area and pre-process the data, the edge server cannot decrypt the $c_{ij}$ without the private key. The edge servers can only aggregate local data directly and transmit the local data aggregated result to the master node. Thus, even if the attacker succeeds in breaking into the edge server, he cannot read any private information about the edge server. On the other hand, when the control center reads the related information in the blockchain, which recovers it as the sum of each smart meter's data in the same area $(D_1, D_2, ..., D_n)$ and store the results of these aggregated plaintext in the local database. Even if some attackers succeed in breaking into the control center, the attackers still cannot obtain the individual power consumption $d_{ij}$. Based on the above analysis, the confidentiality and privacy of each entity in EBDA can be protected.

### B. Authentication and Integrity

In our EBDA scheme, to authenticate the data source for each slot, the one-way hash chain technology is used in this work. Each smart meter has a unique $HC_j^i$, which is generated by TA. For each smart meter $SM_{ij}$, the hash value $h_{ij\_(s-1)}$ was verified in $T_{s-1}$. According to

$$h_{ij\_(s-1)} = h(h_{ij\_s} || T_s) \tag{35}$$

, we can authenticate the validity of $h_{ij\_s}$. Since the hash function is unidirectional, it is difficult to compute $h_{ij\_s}$ from $h_{ij\_(s-1)}$. Therefore, we can verify $h_{ij\_s}$ to determine the validity of the data source. Once $h_{ij\_s}$ validation fails, the edge server assumes that the transmission has been hacked by an external attacker and denies this call.

The BLS [28] short signature and blockchain technology are used to guarantee data integrity. When any edge server $ES_i$ completes the first data aggregation of its area data, it uses the BLS short signature technique to digitally sign the content in the form of (32). Since the BLS short signature is provably secure under the Computational Diffie-Hellman (CDH) problem in the random oracle model [29], the source authentication and data integrity can be guaranteed. After the master node and the candidate node verify the integrity of the block through the distributed voting mechanism, this block is added to the blockchain. Since the blockchain is tamper-proof, the integrity of the data in the blockchain can be guaranteed.

### C. Robustness

If some $SM_{ij}$ break down, $ES_i$ cannot receive $m_i$ data set. This scenario prevents the edge server from getting the correct data format, affecting the decryption of the data in the control center. Here, we denote $U_i$ as the set of all smart meters in the area $i$ and $U_i^{'}$ as the set of failed smart meters in the area $i$ $(U_i^{'} \subset U_i)$. When $ES_i$ collects data at time slot $T_s$, $ES_i$ adds all faulty smart meters to the set $U_i^{'}$. Then, $ES_i$ computes

$$H_1(T_s)^{'} = H_1(T_s)^{\sum\limits_{j \in U_i^{'}} \phi_{ij}} \tag{36}$$

and performs the following local data aggregation operation:

$$\begin{aligned}C_i^{'} &= H_1(T_s)^{'} \prod_{j\in {}^{U_i}/_{U_i^{'}}} c_{ij}\cdot H_1(T_s)^{\phi_{i0}} \bmod N^2\\ &= H_1(T_s)^{\sum_{j\in U_i^{'}}\phi_{ij}}\cdot \prod_{j\in {}^{U_i}/_{U_i^{'}}} g_i^{d_{ij}}\cdot H_1(T_s)^{\phi_{ij}}\cdot H_1(T_s)^{\phi_{i0}} \bmod N^2\\ &= \Big(\prod_{j\in {}^{U_i}/_{U_i^{'}}} g_i^{d_{ij}}\Big)\cdot H_1(T_s)^{\sum_{j\in U_i^{'}}\phi_{ij}+\sum_{j\in {}^{U_i}/_{U_i^{'}}}\phi_{ij}+\phi_{i0}} \bmod N^2\\ &= g_i^{\sum_{j\in {}^{U_i}/_{U_i^{'}}} d_{ij}}\cdot H_1(T_s)^{\sum_{j=0}^{m_i}\phi_{ij}} \bmod N^2\end{aligned} \tag{37}$$

Therefore, even if some smart meters fail, our scheme can still provide the correct data aggregation results.

## VII. Performance Evaluation

In this section, we evaluate the efficiency of our EBDA scheme in terms of the security performance, the computation cost, and the communication overhead. We first compare the EBDA scheme with the traditional scheme, which do not integrate with the blockchain. The traditional scheme represents the data aggregation scheme without edge blockchain assistance in three-tier architecture smart grid system. Compared with the traditional scheme, EBDA has a stronger adaptability and ability to resist attacks. In other words, when encountering the same attack, EBDA is less likely to be attacked for various deployment environments. Then, we compare our EBDA scheme with two existing data aggregation schemes, EPPA [30] and LPDA-EC [14], both of which are designed based on homomorphic encryption scheme. Specifically, simulations are performed to demonstrate the actual efficiency of EBDA. The simulations are conducted on the machine with Intel Core i5-8250U CPU @ 1.80 GHZ and 8.00 GB RAM.

### A. Security Performance

In this subsection, we analyze the security performance of the EBDA scheme. We compute the probability that the attacker successfully launches the forgery data injection attack in the edge layer. We compare the EBDA scheme with the traditional scheme through both theoretical analysis and simulations.

Since our goal is to improve the security performance of edge layer through the introduction of blockchain, we only consider the case of attacking against the edge layer. We consider three attack methods and calculate the average probability of a successful attack in each scheme. For both schemes, the same preconditions are applied: (1) attack targets are limited to edge servers and control center; (2) if the attacker wants to launch a successful forgery data injection attack, which is also called network attack, he needs to control at least $f(f \leq n)$ servers; (3) all attack methods are independent.

For the traditional scheme, a cyber attacker may launch an attack in three ways: (1) attack edge servers and forge data in the data aggregation stage; (2) intercept packets in the channel and tamper with the information; and (3) attack the control center to modify the contents of the database.

Suppose that in the first case of traditional scheme, the probability of the attacker hijacks each edge server is $(\lambda_1, \lambda_2, ..., \lambda_i, ..., \lambda_n)$ for every edge server, where $0 \leq \lambda_i \leq 1; i = 1, 2, ..., n$. The attacker needs to hack into corresponding $f$ edge servers, with probability

$$P_{a_1} = \prod_{i=1}^{f} \lambda_i \tag{38}$$

Suppose that in the second case of traditional scheme, the probability of the attacker successfully invading the channel between the edge server and the control center is $(\eta_1, \eta_2, ..., \eta_i, ..., \eta_n)$, where $0 \leq \eta_i \leq 1; i = 1, 2, 3, ..., n$. Since the attacker needs to hijack $f$ corresponding communication channels, the probability of a successful attack is:

$$P_{a_2} = \prod_{i=1}^{f} \eta_i \tag{39}$$

For the last situation, we use $\mu$ to refer to the probability that an attacker successfully hijacks the control center and modify the data, where $0 \leq \mu \leq 0.1$. Finally, the probability $P_a$ that an attacker successfully launches a cyber-attack in traditional scheme can be calculated as:

$$P_a = \frac{1}{3}\Big(\prod_{i=1}^{f} \lambda_i + \prod_{i=1}^{f} \eta_i + \mu\Big) \tag{40}$$

For the EBDA scheme, a cyber attacker may launch an attack in three ways: (1) attack edge servers and forge data in the data aggregation; (2) intercept packets between the master node and the remaining candidates nodes, and tamper with the information; (3) attackers steal the identity of the candidate node and tamper with the voting results. Note that we don't consider the situation that the control center is attacked in EBDA since no data is stored in the control center in EBDA.

Next, we analyze the security performance of EBDA. The probability of the attacker stealing the key of each server is $(\beta_1, \beta_2, ..., \beta_i, ..., \beta_n)$, where $0 \leq \beta_i \leq 1; i = 1, 2, 3, ..., n$. We use $(\overline{\lambda_1}, \overline{\lambda_2}, ..., \overline{\lambda_i}, ..., \overline{\lambda_n})$ represents the probability of hijacking the corresponding edge server, where $0 \leq \overline{\lambda_i} \leq 1; i = 1, 2, ..., n$. The attacker must hijack $f$ edge servers and obtain the corresponding secret key information to encrypt the forged data information. Therefore, in this case, the probability of success is:

$$P_{b_1} = \Big(\prod_{i=1}^{f} \overline{\lambda_i}\Big)\cdot\Big(\prod_{i=1}^{f} \beta_i\Big) \tag{41}$$

For the second situation in EBDA, there are $w$ candidate nodes in the edge layer for maintaining the blockchain together, so the number of communication channels between the master node and the remaining candidate nodes during the consensus stage is $w(w-1)/2$ . We assume that the probability for attackers to tamper with the block when transmitting from one node to another is denoted as $(\overline{\eta_1}, \overline{\eta_2}, ..., \overline{\eta_{w(w-1)/2}})$, where $0 \leq \overline{\eta_i} \leq 1; i = 1, 2, ..., w(w-1)/2$ . The attacker must hijack at least

$$\overline{k} = ceil[\tau \cdot \frac{w(w-1)}{2}] \tag{42}$$

communication channels in order to launch a successful attack, where $\tau$ is the voting threshold. In addition, the attacker must also obtain the key of $f$ nodes. Therefore, the success probability of launching this attack for this situation is:

$$P_{b_2} = (\prod_{i=1}^{\overline{k}} \overline{\eta_i}) \cdot (\prod_{i=1}^{f} \beta_i) \tag{43}$$

For the third situation in EBDA, the attacker tries to steal the identity of the candidate node and tamper with the voting results. The attacker also needs to hijack at least $k = ceil(\tau \cdot w)$ candidate nodes and obtain the keys of $f$ nodes to complete a successful attack. In this situation, the succcess probability of the attacker is:

$$P_{b_3} = (\prod_{i=1}^{k} \overline{\lambda_i}) \cdot (\prod_{i=1}^{f} \beta_i) \tag{44}$$

Therefore, the overall successful probability, $P_b$, of launching this attack for EBDA, can be computed as:

$$P_b = \frac{1}{3}[(\prod_{i=1}^{f} \overline{\lambda_i}) \cdot (\prod_{i=1}^{f} \beta_i) + (\prod_{i=1}^{\overline{k}} \overline{\eta_i}) \cdot (\prod_{i=1}^{f} \beta_i) + (\prod_{i=1}^{k} \overline{\lambda_i}) \cdot (\prod_{i=1}^{f} \beta_i)] \tag{45}$$

The above performances against attacks are summarized in Table III.

TABLE III
SUCCESSFUL PROBABILITY

| Schemes | Successful probability |
|---|---|
| Traditional scheme | $\frac{1}{3}(\prod_{i=1}^{f} \lambda_i + \prod_{i=1}^{f} \eta_i + \mu)$ |
| EBDA | $\frac{1}{3}[(\prod_{i=1}^{f} \overline{\lambda_i}) \cdot (\prod_{i=1}^{f} \beta_i) + (\prod_{i=1}^{\overline{k}} \overline{\eta_i}) \cdot (\prod_{i=1}^{f} \beta_i) + (\prod_{i=1}^{k} \overline{\lambda_i}) \cdot (\prod_{i=1}^{f} \beta_i)]$ |

According to the above analysis, we give the mathematical expression of the attacker successfully launching a network attack. By comparing (45) and (40), we can find that (45) is less than (40). However, since there are many variables in each function, we cannot directly and quantitatively measure the difference. Therefore, we introduce the Monte Carlo method. We use the Monte Carlo method to simulate two schemes. We consider that the edge layer has 100 edge servers and the control center has a cloud server. Then, we consider that the percentage of edge servers that the attacker needs to manipulate ranges from 10% to 100%; thus, the parameter $f$ changes from 10 to 100. Meanwhile, all values of $(\lambda_i, \eta_i, \overline{\lambda_i}, \beta_i, \overline{\eta_i})$ are randomly distributed in [0.9,1], the value of $\mu$ is in [0,0.1], and the range of $\tau$ is [0.5,1]. In each simulation execution, the values of these parameters are randomly selected. The value of $w$ is in [6, 100]. The number of simulations is defined as 1000.

The comparison depicted in Fig. 8 clearly shows the advantages of the EBDA scheme. When attacker hijacks the same number of edge servers, the probability of EBDA scheme being attacked successfully is much lower than the traditional scheme. Most importantly, we can see that the security performance of EBDA scheme is significantly improved with the increasing of the number of candidate nodes. Therefore, we can dynamically adjust the number of candidate nodes according to the deployment environment of smart grid.

## *B. Computation Cost*

In this subsection, we analyze the computation cost of the whole system. We assume that the number of edge servers changes from 5 to 50 one by one, and each edge server is responsible for 20 smart meters. To measure the performance of the proposed scheme, we compared EBDA with two existing schemes: EPPA and LPDA-EC. Both schemes are based on the homomorphic encryption algorithm, which is similar to our scheme. We define the RSA modules $N$ and the parameter $p_0$ as 1024 bits and 160 bits, respectively. For convenience, we define $T_{E_1}, T_{E_2}, T_M, T_P$ as the exponentiation operations in $Z_{N^2}^*$, the exponential operations in $G$, the multiplication operations, and the bilinear pairing in $G$, respectively. We use the Type A curve of Pairing-Based Cryptography (PBC) library [31] to implement these operations. Table IV lists the operations and the corresponding running time. Note that, since the cost of hash operation is negligible compared with exponential operation and multiplication operation, the computation cost of hash operation is not considered in our evaluation.

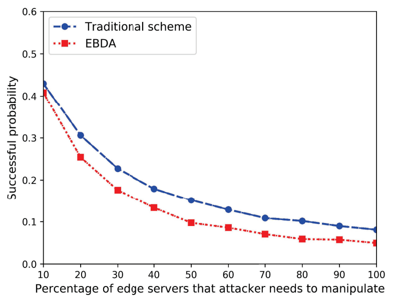
TABLE IV
RUNNING TIME OF OPERATIONS

| Notations | Descriptions | Time cost(ms) |
|---|---|---|
| $T_{E_1}$ | Exponentiation Operation in $Z_{N^2}^*$ | 1.60 |
| $T_{E_2}$ | Exponentiation Operation in $G$ | 1.62 |
| $T_M$ | Multiplication Operation | 0.06 |
| $T_P$ | Pairing Operation | 17.70 |

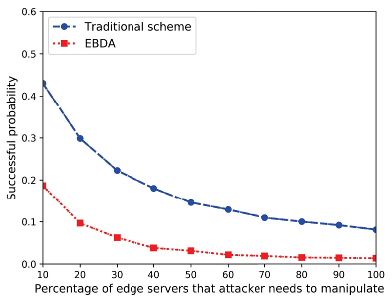
In the user layer, the generation of ciphertext $c_{ij}$ requires one exponentiation operation $T_{E_1}$ in $Z_{N^2}^*$ and one multiplication operation $T_M$, respectively. When the edge server receives the data, it begins to perform the local data aggregation operation to generate the aggregation ciphertext $C_i$, which requires $m_i + 1$ multiplication operations $T_M$. Then the edge server performs digital signature, which requires one multiplication operation $T_M$. When the edge server transmits the transaction to the master node, the master node needs to validate $n-1$ transactions, which requires $n+1$ pairing operation $T_P$. When the data validation is complete, the master node begins the global data aggregation operation to generate the ciphertext $C$, which requires $n$ multiplication operations $T_M$. When the control center consults the blockchain content, it decrypts the ciphertext with its private key, by computing equation (33), which requires one exponentiation operation $T_{E_1}$ in $Z_{N^2}^*$ and one multiplication operations $T_M$. To sum up, the overall computational cost of EBDA is

$$\sum_{i=1}^{n} [m_i T_{E_1} + 2(m_i+1)T_M] + (n+1)T_p + (n+1)T_M + T_{E_1} \tag{46}$$

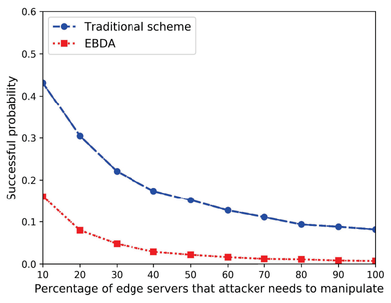
Similarly, we compute the computation cost of EPPA scheme and LPDA-EC scheme, as shown in Table V. Although the information in Table V cannot be accurately analyzed, combined with Table IV, we find that the computation cost of EBDA is significantly lower than EPPA and LPDA-EC.



(a) 6 candidate nodes in EBDA



(b) 50 candidate nodes in EBDA



(c) 100 candidate nodes in EBDA

Fig. 8. Successful attack probability

TABLE V
COMPUTATION COST

| Schemes | Computation cost |
|---|---|
| EPPA | $\sum_{i=1}^{n} [m_iT_{E_1} + (m_i+1)T_M + (m_i+3)T_p] + 2nT_p + nT_{E_1}$ |
| LPDA-EC | $\sum_{i=1}^{n} [2m_iT_{E_1} + 4m_iT_M + (3m_i+1)T_{E_2}] + 2nT_p + 2nT_{E_1}$ |
| EBDA | $\sum_{i=1}^{n} [m_iT_{E_1} + 2(m_i+1)T_M] + (n+1)T_p + (n+1)T_M + T_{E_1}$ |

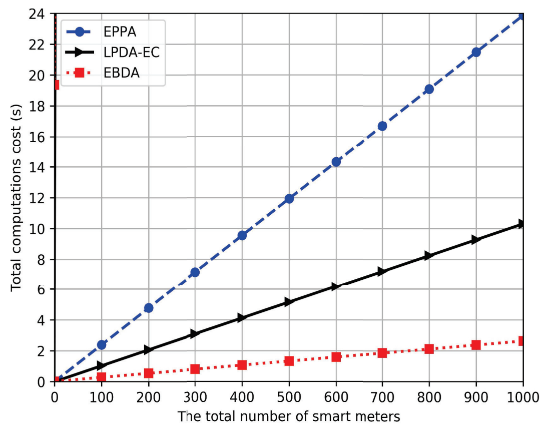


Fig. 9. Computation cost

We can see from Fig. 9 that EBDA shows more superiority when the number of smart meters increases. For examples, when the number of smart meters is 800, the total computation cost of EBDA is around 2s, which reduces by 90% and 75% that of EPPA and LPDA-EC, respectively. We can conclude that EBDA can significantly reduce the computation cost, and shows great expansibility.

### C. Communication Overhead

The communication overhead of EBDA consists of communication overhead between smart meter and edge server, and communication overhead between edge server and master node. For the convenience, we only consider the communication overhead of one region. First, the smart meter generates its own data report and sends it in $(c_{ij}, h_{ij\_s}, mac_{ijs})$ to the edge server. The size is $S_{user-edge} = |c_{ij}| + |h_{ij\_s}| + |mac_{ijs}| =$ 2368 bits if we define $|N^2| = 2048$ bits, $|p_0| = 128$ bits. Therefore, at each time slot, the overall communication overhead between smart meters and edge server is $S_E = m_i \cdot S_{user-edge}$.

Next, we analyze the communication overhead between edge server and master node, the edge server performs the local data aggregation, aggregates $m_i$ encrypted ciphertexts, generate a transaction package $(C_i, ES_i, \sigma_i)$, and then transmit the transaction to the master node. After performing the local data aggregation, the communication overhead in the system can be significantly reduced, reducing from the original $S_E$ to $S_{edge-master} = |C_i| + |ES_i| + |\sigma_i| = 2368$ bits. This means that the communication overhead is independent of the number of smart meters. We conclude the above communication overheads in Table VI.

TABLE VI
COMMUNICATION OVERHEAD

| Schemes | Communication overhead |
|---|---|
| EPPA | $m_i(160 + \|c_{ij}\| + T_s + 160 + 160) + \|ES_i\| + \|C_i\| + T_s + 160 + \|\sigma_i\|$ |
| LPDA-EC | $m_i(160 + \|c_{ij}\| + T_s + 160) + \|ES_i\| + \|C_i\| + T_s + \|\sigma_i\|$ |
| EBDA | $m_i(\|c_{ij}\| + \|h_{ij\_s}\| + \|mac_{ijs}\|) + \|C_i\| + \|ES_i\| + \|\sigma_i\|$ |

We depict the communication overhead of each scheme in Fig. 10. From Fig. 10, we can see that compared with EPPA and LPDA-EC, the proposed scheme is more effective in communication overhead, so that communication resources can be used efficiently.

In conclusion, EBDA shows great superiority in terms of security performance, computation cost, and communication overhead, comparing with the existing schemes. EBDA is a lightweight, low cost, and low risk data aggregation scheme.

## VIII. CONCLUSION

In this paper, we have proposed an edge blockchain assisted lightweight privacy-preserving data aggregation for smart grid

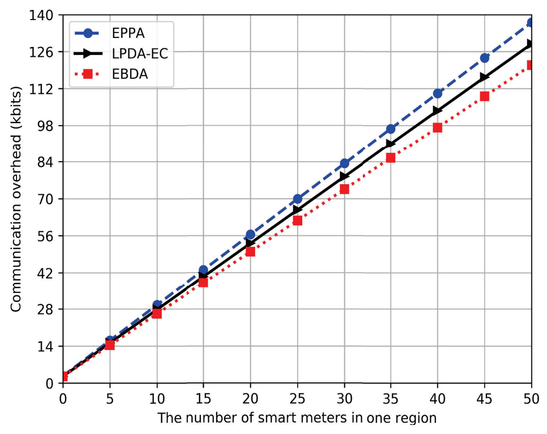


Fig. 10. Communication overheads

(EBDA). EBDA can not only filter the forgery data at the user layer in advance, but also be robust. Even if some smart meters are offline, the local data aggregation operation can still be completed. EBDA not only integrates edge computing, but also incorporates blockchain technology. When the master node completes the global data aggregation, the aggregated results will be packaged into transactions and stored in the edge blockchain. The edge layer takes advantage of the decentralized and tamper-proof features of the blockchain, and successfully enhances the system's ability to resist network attacks. Through theoretical analysis and simulation, we show that the proposed EBDA scheme is secure and reliable. In addition, through the analysis of computation cost and communication overhead, we show that the designed scheme is a lightweight data aggregation scheme.

## References

[1] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward intelligent machine-to-machine communications in smart grid," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 60–65, 2011.

[2] H. Liang, B. J. Choi, W. Zhuang, and X. Shen, "Towards optimal energy store-carry-and-deliver for phevs via v2g system," in *Proc. IEEE Conf. Comput. Commun.*, Mar. 2012, pp. 25–30.

[3] W. Ketter, J. Collins, M. Saar-Tsechansky, and O. Marom, "Information systems for a smart electricity grid: Emerging challenges and opportunities," *ACM Transactions on Management Information Systems (TMIS)*, vol. 9, no. 3, pp. 1–22, 2018.

[4] S. Zahoor, N. Javaid, A. Khan, B. Ruqia, F. J. Muhammad, and M. Zahid, "A cloud-fog-based smart grid model for efficient resource utilization," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 1154–1160.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[6] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb), Washington, DC, USA*, Nov. 2015, pp. 73–78.

[7] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proc. 12th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2014, pp. 68–81.

[8] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic execution between mobile device and cloud," in *Proc. 6th Conf. Comput. Syst.*, 2011, pp. 301–314.

[9] R. Deng, R. Lu, C. Lai, and T. H. Luan, "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing," in *Proc. IEEE Int. Conf. Commun. (ICC), London, U.K.*, Jun. 2015, pp. 3909–3914.

[10] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 10, pp. 2991–3005, 2016.

[11] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, no. 1, pp. 680–698, 2018.

[12] L. Shen, J. Ma, X. Liu, F. Wei, and M. Miao, "A secure and efficient id-based aggregate signature scheme for wireless sensor networks," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 546–554, Apr. 2017.

[13] Z. Wang, "An identity-based data aggregation protocol for the smart grid," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2428–2435, 2017.

[14] J. Zhang, Y. Zhao, J. Wu, and B. Chen, "Lpda-ec: A lightweight privacy-preserving data aggregation scheme for edge computing," in *Proc. IEEE 15th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2018, pp. 98–106.

[15] J. Xu, Y. Wu, X. Luo, and D. Yang, "Improving the efficiency of blockchain applications with smart contract based cyber-insurance," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–7.

[16] G. Xue, J. Xu, H. Wu, W. Lu, and L. Xu, "Incentive mechanism for rational miners in bitcoin mining pool," *Information Systems Frontiers*, 2020. [Online]. Available: https://doi.org/10.1007/s10796-020-10019-2

[17] Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani, X. Du, and Y. Ma, "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 82–88, 2018.

[18] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 840–852, 2018.

[19] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Distributed blockchain-based data protection framework for modern power systems against cyber attacks," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3162–3173, 2018.

[20] 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, 2018.

[21] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain

and edge computing systems: A survey, some research issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1508–1532, 2019.

[22] A. Shahraki, A. Taherkordi, . Haugen, and F. Eliassen, "A survey and future directions on clustering: From wsns to iot and modern networking paradigms," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2020, doi: 10.1109/TNSM.2020.3035315.

[23] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2536–2549, 2020.

[24] R. Casado-Vara, F. de la Prieta, J. Prieto, and J. M. Corchado, "Blockchain framework for iot data quality via edge computing," in *Proc. 1st Workshop Blockchain-Enabled Netw. Sensor Syst., Shenzhen, China*, Nov. 2018, pp. 19–24.

[25] A. Stanciu, "Blockchain based distributed control system for edge computing," in *Proc. 21st Int. Conf. Control Syst. Comput. Sci. (CSCS)*, May. 2017, pp. 667–671.

[26] X. Wang, S. Garg, H. Lin, G. Kaddoum, J. Hu, and M. S. Hossain, "A secure data aggregation strategy in edge computing and blockchain empowered internet of things," *IEEE Internet of Things Journal*, pp. 1–1, 2020, doi: 10.1109/JIOT.2020.3023588.

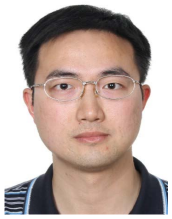
[27] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. TheoryAppl. Cryptogr. Tech. Adv. Cryptol. (EUROCRYPT), Prague, Czech Republic*, J. Stern, Ed., May, 1999, pp. 223–238.

[28] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *J. Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[29] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. ACM Conf. Computer and Comm. Security*, 1993, pp. 62–73.

[30] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[31] B. Lynn, "The pairing-based cryptography (pbc) library," Jun. 2013. [Online]. Available: http://crypto.stanford.edu/pbc



**Weifeng Lu** is currently an Associate Professor at Nanjing University of Posts and Telecommunications. He received his Ph.D. in information security from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2008. He finished his Postdoctoral research work in Information Science and Engineering of Southeast University in 2016. His research interest is in radio resource management in device-to-device networks and edge computing, blockchain, etc.

**Zhihao Ren** received the B.E. degree in Computer Science and Technology from School of Computer Science, Yancheng Institute of Technology, China, in 2018. Now he is currently pursuing the M.E. degree at Nanjing University of posts and telecommunication, China. His research interest is the data aggregation of edge computing.

**Jia Xu** (M'15) received the M.S. degree in School of Information and Engineering from Yangzhou University, Jiangsu, China, in 2006 and the PhD. Degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in the School of Computer Science at Nanjing University of Posts and Telecommunications. He was a visiting Scholar in the Department of Electrical Engineering & Computer Science at Colorado School of Mines from Nov. 2014 to May. 2015. His main research interests include crowdsourcing, edge computing and wireless sensor networks. Prof. Xu has served as the PC Co-Chair of SciSec 2019, Organizing Chair of ISKE 2017, TPC member of Globecom, ICC, MASS, ICNC, EDGE. He currently serves as the Publicity Co-Chair of SciSec 2021.

**Siguang Chen** (M'17) is currently an Associate Professor at Nanjing University of Posts and Telecommunications. He received his Ph.D. in information security from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2011. He finished his Postdoctoral research work in City University of Hong Kong in 2012. From 2014 to 2015, he also was a Postdoctoral Fellow in the University of British Columbia. He has published more than 70 papers and applied 30 patents, serves as Editor of EAI Endorsed Transactions on Cloud Systems, Editor of Journal on Internet of Things, Corresponding Experts of Engineering Journal, and serves as General Co-Chair of ICAIS/ICCCS 2019-2021 Workshop on Mobile, Wireless and Sensors Networking. He also served/serves as a TPC member in IOP 2015, WCSP 2016, ICCT 2017, 2018 & 2019, CISIS 2018, ICCCS 2018 Workshop, GLOBECOM 2018 Workshop, ICC 2019 & 2020, ICCC 2019 & 2020 and IEEE EDGE 2020. His current research interests are in the area of fog/edge computing, deep/reinforcement learning, privacy preserving of big data and IoT resource optimization.