Engineering 8 (2022) 159–169

Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
Smart Grid and Energy Internet—Article

Double-Blockchain Assisted Secure and Anonymous Data Aggregation for Fog-Enabled Smart Grid

Siguang Chen [a,b,*], Li Yang [a,b], Chuanxin Zhao [c], Vijayakumar Varadarajan [d], Kun Wang [e]

[a] Jiangsu Key Lab of Broadband Wireless Communication and Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[b] Jiangsu Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[c] Anhui Provincial Key Laboratory of Network and Information Security, Anhui Normal University, Wuhu 241000, China
[d] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
[e] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095, USA

ARTICLE INFO

Article history:
Received 20 August 2019
Revised 11 May 2020
Accepted 27 June 2020
Available online 11 August 2020

Keywords:
Blockchain
Fog computing
Homomorphic encryption
Smart grid
Anonymity

ABSTRACT

As a future energy system, the smart grid is designed to improve the efficiency of traditional power systems while providing more stable and reliable services. However, this efficient and reliable service relies on collecting and analyzing users' electricity consumption data frequently, which induces various security and privacy threats. To address these challenges, we propose a double-blockchain assisted secure and anonymous data aggregation scheme for fog-enabled smart grid named DA-SADA. Specifically, we design a three-tier architecture-based data aggregation framework by integrating fog computing and the blockchain, which provides strong support for achieving efficient and secure data collection in smart grids. Subsequently, we develop a secure and anonymous data aggregation mechanism with low computational overhead by jointly leveraging the Paillier encryption, batch aggregation signature and anonymous authentication. In particular, the system achieves fine-grained data aggregation and provides effective support for power dispatching and price adjustment by the designed double-blockchain and two-level data aggregation. Finally, the superiority of the proposed scheme is illustrated by a series of security and computation cost analyses.

1. Introduction

Smart grids, as next-generation power networks, provide efficient and intelligent electricity and information exchange to maximize energy usage efficiency and meet modern demands by integrating advanced information processing and communication technologies [1,2]. For example, the smart meter in a user's home can sense the electricity usage information of home appliances in real time, and the control center can collect and analyze these data to learn the user's power usage behaviors and provide dynamic pricing and flexible power dispatching policies [3–5]. However, the smart grid is confronted with substantial data communication and computation burdens given the explosive growth of smart meters [6,7]. Furthermore, the exposure of the collected power consumption data of smart meters promotes privacy leakage risks because the power consumption data can be used to explore users' living habits and even infer their economic status [8]. In addition, tampering and forgery attacks will also produce a great threat to the stability of the smart grid [9,10]. For example, the false data injection attack from a cyber attacker caused the world-shaking Ukraine blackout accident in 2015 [11]. To address the above challenges of performance, privacy, and security in the smart grid, many research schemes were proposed, among which a typical representative, that is a secure and efficient data aggregation mechanism, has attracted appreciable attention for its significant advantage. Currently, the smart grid data aggregation schemes can be roughly divided into the following three categories.

The first category is composed of the data aggregation scheme with the traditional network architecture. For example, Lu et al. [12] presented an efficient and privacy-preserving data aggregation mechanism by integrating the superincreasing sequence, homomorphic Paillier encryption, and batch verification, achieving efficient multidimensional data aggregation with security and privacy protections. Furthermore, Ni et al. [13] constructed a

[*] Corresponding author.
E-mail address: sgchen@njupt.edu.cn (S. Chen).

https://doi.org/10.1016/j.eng.2020.06.018
2095-8099/

security-enhanced data aggregation scheme by jointly using homomorphic encryption, the trapdoor hash function, and homomorphic authenticators, thereby improving the computation and communication costs of work with confidentiality and integrity guarantees. From the perspective of dynamic pricing and service support, Gope and Sikdar [14] formulated a privacy-friendly lightweight data aggregation mechanism. It realizes strong privacy protection under dynamic billing, which is especially suitable for devices with limited computing resources. Without the support of a trusted third party, Liu et al. [15] proposed a practical data aggregation scheme with efficient privacy preservation. In the proposed scheme, the trusted users are linked to form a virtual aggregated area, and the aggregated results are used for data analysis, such that the user's personal privacy is protected and the robustness of the system is improved. From the perspective of fine-grained aggregation, Li et al. [16] developed a multisubset data aggregation scheme with efficient privacy preservation. According to the different ranges of the power consumption data, it can achieve multisubset aggregation and provide fine-grained data service; at the same time, the user's privacy is preserved with a low computation cost. Although the developed schemes in the above literature achieve efficient and secure data aggregation, further opportunities exist to reduce the data processing delay and communication overhead that are due to the weakness of the employed traditional network architecture.

Fortunately, fog computing, as a promising computing paradigm, has been developed to overcome the weakness of the traditional network architecture and has been proven to decrease the delay and communication overhead significantly, especially when combined with cloud computing [17]. Consequently, the second category of solutions developed the data aggregation mechanism with the edge/fog computing architecture. For example, Lu et al. [18] constructed a fog-assisted privacy-preserving data aggregation scheme by integrating the Paillier encryption, the one-way hash chain, and Chinese remainder theorem. This scheme has the property of aggregating the data of hybrid Internet of Things (IoTs) into one, and possesses a filtering function for fake data. Based on the application demands of different data types, Huang et al. [19] studied a fog-enabled selective data aggregation scheme that also considers the reliability and privacy-preserving problems. To further enhance the privacy effect of the above methods, Lyu et al. [20] proposed a fog-based differential privacy-preserving data aggregation scheme; this scheme achieves differential privacy for statistical data and ensures the data confidentiality from the aggregator. From the resource-constraint consideration in an edge computing system, Zhang et al. [21] presented an efficiency-enhanced privacy-preserving data aggregation scheme by transferring the time-consuming signature operations offline, thereby effectively relieving the online computation burden. Focusing on anonymous authentication in the fog-enabled smart grid, Zhu et al. [22] conceived an anonymous data aggregation scheme by employing the Paillier cryptosystem and blind signature, which can provide strong privacy protection with low computation and communication costs. Although the above solution reduces the system delay and communication overheads significantly, and provides privacy and security protection to some degree, this category of schemes still faces issues of security and centralization. For example, when a user's private information is transmitted to a fog node, and a malicious attacker successfully intercepts the channel and steals the secret key, it is difficult to guarantee the privacy of the user. Moreover, all of the users' data are concentrated in the fog or cloud layer, which inevitably introduces the problem of centralization.

The emergence of the blockchain technique [23] has provided a new perspective to address the above problems because of its decentralization and nontampering features. Currently, there are several studies that have applied the blockchain to the smart grid. For example, in Ref, [24], Liang et al. investigated a blockchain-based data protection scheme for the smart grid, and it proves that the blockchain can effectively improve system security under cyber-attacks. Therefore, the third category of solutions encompasses the combination of data aggregation and the blockchain technique. Specifically, Fan and Zhang [25] proposed a secure data aggregation for smart power regulation by integrating the consortium blockchain into the smart grid, in which a multireceiver model for collecting multidimensional data is developed, and based on smart contracts, it establishes flexible power monitoring and management mechanisms to enhance the security of the smart grid. Guan et al. [26] studied a blockchain-assisted anonymous data aggregation scheme for the smart grid; it enhances the system security and obtains better performance compared with other solutions. However, the users' power consumption data are transmitted in plaintext form in groups and will be confronted with some security risks. Although the above blockchain-based privacy-preserving data aggregation schemes effectively enhance the smart grid security and solve the problem of centralization and single point failure, all of them do not consider the edge computing paradigm, causing an ineffective utilization of local resources. As a result, the system efficiency has a large space for improvement. Accordingly, the works [27] and [28] were developed to improve system performance by combining the blockchain and edge computing, but these two schemes do not provide specific executable solutions.

The above schemes solve the corresponding problems of the smart grid to varying degrees, but there are still many weaknesses. Different from the existing solutions, we propose a double-blockchain assisted secure and anonymous data aggregation (DA-SADA) scheme for the fog-enabled smart grid by integrating the blockchain, the Paillier cryptosystem, batch verification, and an anonymous authentication mechanism. Specifically, the main contributions of this scheme are summarized as follows:

(1) We design a three-tier architecture-based data aggregation framework by integrating fog computing and the blockchain. It is a security-enhanced framework, and the local resources are exploited effectively, which provides strong support for achieving efficient and secure data collection in the smart grid.

(2) We develop a secure and anonymous data aggregation mechanism with low computational overhead by jointly leveraging the Paillier encryption, batch aggregation signature, and anonymous authentication. It can effectively resist various security threats (such as eavesdropping, tampering, and replay attacks) and provide multiple privacy preservations.

(3) The system achieves fine-grained data aggregation and provides effective support for power dispatching and price adjustment by the designed double-blockchain and two-level data aggregation. Additionally, this design further strengthens the system security and robustness.

The remaining parts of this paper are organized as follows. In Section 2, we describe some preliminaries. Section 3 introduces the constructed network model in detail. Our proposed scheme is presented in Section 4, followed by the security and performance evaluation in Section 5. Section 6 ultimately draws the conclusion of this paper.

2. Preliminaries

2.1. Blockchain

The blockchain can be considered as a peer to peer (P2P) distributed database that creates blocks and links in chronological order [29], which is designed to provide decentralized and distributed solutions for a wide range IoT and industrial Internet of

Things (IIoT) applications. The main blockchain components include transactions, blocks, smart contracts, the consensus mechanism, cryptography, and the P2P network [30]. Specifically, in a blockchain network, the participants act as the distributed nodes for protecting and maintaining the shared record of transactions collaboratively; it does not need any trusted party for supervision and management. All nodes are responsible for sharing, packaging, verifying, and storing new transactions generated in the blockchain network. Therefore, it can establish trust among participating entities that do not trust each other in a distributed scenario. It also has decentralization, nontampering, and security features.

Decentralization: The distributed structure of the blockchain ensures the decentralization property. Furthermore, the third-party maintenance management is not required, and the nodes in the network are completely autonomous based on the incentive mechanism.

Nontampering: Nontampering means that once transaction data are recorded in the blockchain, the record cannot be successfully tampered with or deleted.

Security: The data written to the blockchain needs to be collectively verified, which indicates that successful tampering needs at least 51% of the computing power in the entire network, which is usually impossible in practice.

2.2. Paillier encryption

The Paillier homomorphic encryption method is widely used in the privacy protection area. It can directly operate on ciphertext, thus effectively protecting data privacy. Specifically, the Paillier encryption is an additive homomorphic encryption, and it consists of key generation, encryption operation, and decryption operation.

Key generation: Given a security parameter κ, the system randomly chooses two large primes p and q, where $|p| = |q| = \kappa$ (this operation is used to calculate the length of p and q, and which is equal to κ-bits) and $\gcd[pq,\ (p-1)(q-1)] = 1$, and then calculates public key $N = pq$ and private key $\lambda = \mathrm{lcm}(p-1,\ q-1)$. Next, select a generator $g \in Z_{N^2}^*$ and ensure the calculation of $\mu = \left[L\left(g^{\lambda} \bmod N^2\right)\right]^{-1} \bmod N$ is available. Furthermore, it defines the function $L(u) = (u-1)/N$. Finally, the public key (N, g) and private key (λ, μ) of Paillier encryption are obtained.

Encryption operation: For any plaintext $m \in Z_N$, the system selects the random number $r = Z_N^*$, and then the ciphertext can be calculated as $C = g^m r^N \bmod N^2$.

Decryption operation: According to ciphertext C, it can calculate the plaintext as $m = L\left(C^{\lambda} \bmod N^2\right) \Big/ L\left(g^{\lambda} \bmod N^2\right) \bmod N$.

2.3. Bloom filter

The Bloom filter consists of a long binary vector and a series of random mapping functions; it has the advantages of low computational complexity, high space utilization, and query efficiency. It can quickly confirm whether an element exists in the set.

We assume that there are k hash functions $\{h_1, h_2, \ldots, h_k\}$ and one set with elements $\{x_1, x_2, \ldots, x_{\omega}\}$. These elements are mapped to the corresponding position of the Bloom filter by k uniformly independent hash functions, and the value of the corresponding position is set to 1. The specific operation is shown in Fig. 1.

Element adding: As shown in Fig. 1, we hash the element by k times to obtain k hash values $\{h_1(x_1), h_2(x_1), \ldots, h_k(x_1)\}$, and then based on these values, find the corresponding positions of the Bloom filter. Finally, let values k of the corresponding positions in the Bloom filter be 1.

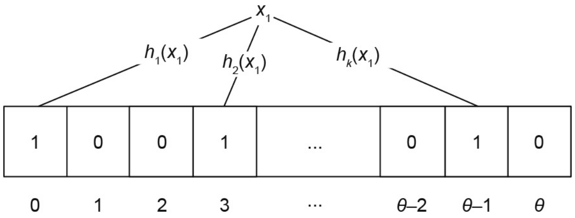

Fig. 1. Generation of the bloom filter.

Element query: To query whether the element x_1 exists in the Bloom filter, we first calculate k hash values of the element x_1, which is denoted as $\{h_1(x_1), h_2(x_1), \ldots, h_k(x_1)\}$, and then check whether the values of the corresponding positions in the Bloom filter are all 1. If one of them is zero, it indicates that the element x_1 is not stored in the Bloom filter; otherwise, the element x_1 is stored in the Bloom filter.

False positive rate: A false positive in terms of the Bloom filter means that the element x_1 does not belong to the element of the Bloom filter, but the value of the corresponding position $\mathrm{BF}[h_i(x)](1 \le i \le k)$ is 1. We assume that the element value in the Bloom filter is set to 1 with probability $p = 1 - (1 - 1/\theta)^{kn}$. According to the result of the work [31], we can obtain that the upper limit of the false positive rate is $\varepsilon = p^k\left\{1 + o\left[k/p\sqrt{\ln(\theta)k\ln(p)/\theta}\right]\right\}$, where θ denotes the number of elements in the Bloom filter.

3. Network model and threats

3.1. Network model

In our constructed network model, a fog-enabled data aggregation smart grid consists of four entities (smart meters, fog nodes, cloud server, and trust authority (TA)) and is displayed in Fig. 2. Specifically, we assume that the coverage area of a smart grid is divided into m subareas, and each subarea deploys n smart meters for sensing user's power consumption information. All of the $m \cdot n$ smart meters form the user layer. Accordingly, each subarea deploys a fog node to collect and aggregate the data from its own area, and all the m fog nodes form the fog computing layer that is located at the edge of the network between the user and service supporting layers. At the service supporting layer, the cloud server is used to process the data uploaded from the fog layer and generate real-time decision-making. TA is responsible for the generation of the entire system's parameters. The specific function definitions of these entities in each layer are presented in detail in the following part.

The User layer: The user layer is mainly composed of a large number of smart meters. For example, in the subarea j, the ith smart meter SM_{ij} observes a user's real-time power consumption, and then encrypts and signs these consumption data. Next, it sends these encrypted data to the aggregation node at the user layer. The aggregation node aggregates the verified ciphertext to generate the first-level aggregation ciphertext, and then encapsulates the related information into a block. At the same time, the newly generated block will be added to the user aggregation (UA)-blockchain by the consensus mechanism. In these processing processes, the identity of SM_{ij} (i.e., the user) always exists under a pseudonym. Finally, the generated UA-blockchain is sent to the fog_j for further processing.

Fog computing layer: The fog computing layer is the middle layer between the user and service supporting layers that is deployed at the edge of the network, which enables the second-level aggregation of the encryption data to significantly reduce the communication overhead. Specifically, when the fog_j receives the first-level aggregated ciphertext from the UA-chain sent by the aggregation node in the user layer, it signs the aggregated

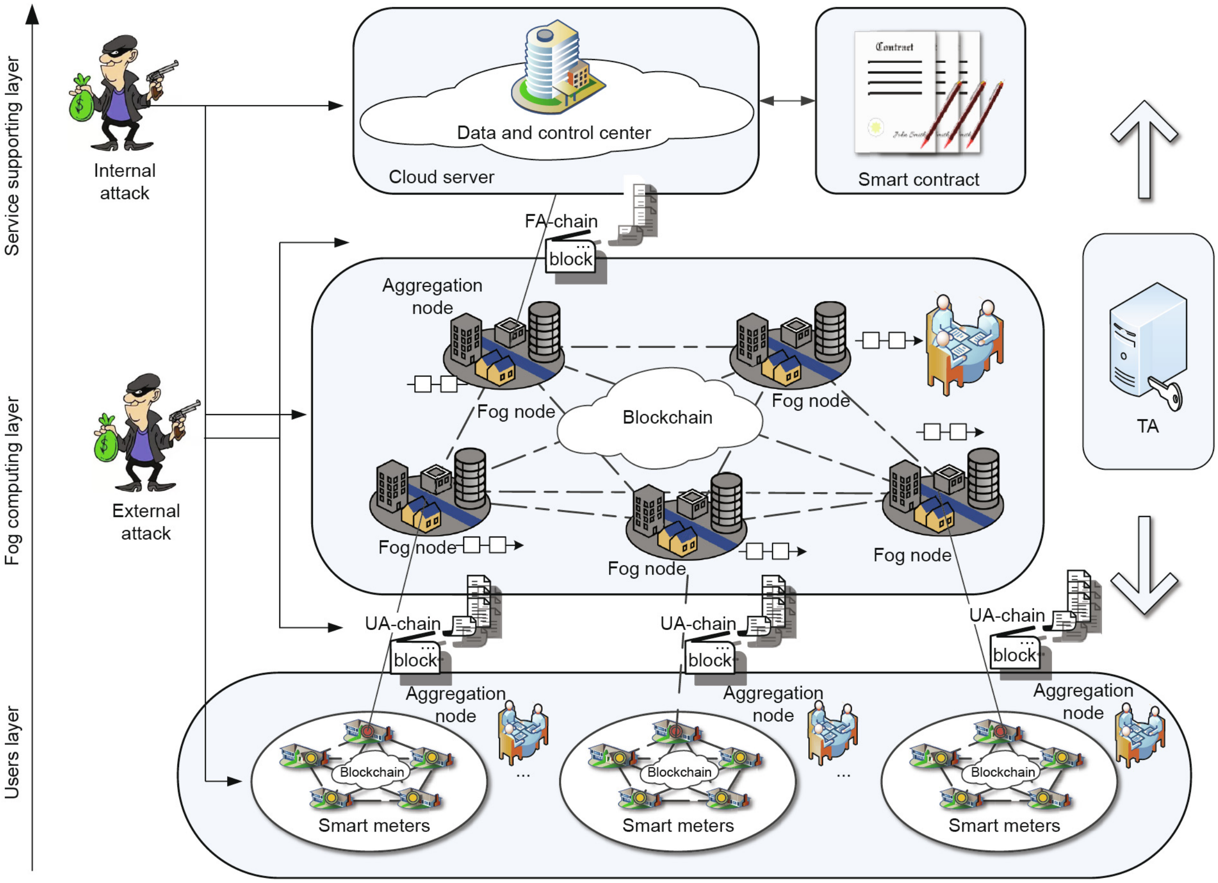

Fig. 2. Network architecture of the developed DA-SADA. UA: user aggregation; FA: fog aggregation.

ciphertext and sends it to the aggregation node at the fog layer for secondary aggregation. Next, the aggregation node encapsulates the related information into a new block, and then the newly generated block is added to the fog aggregation (FA)-blockchain by the consensus mechanism. Finally, the generated FA-chain is sent to the cloud server.

Service supporting layer: In this layer, the cloud server can record, analyze, store, and manage users' power usage information in real time, which is automatically executed by a smart contract, so the whole process does not need human intervention, improving the efficiency of the system and enhancing the security of the privacy data. Specifically, when the cloud server obtains the second-level aggregated ciphertext from the FA-chain that is sent by the aggregation node at the fog layer, it performs the decryption operation to recover the plaintext of the second-level aggregation result, and then utilizes Horner's law to achieve fine-grained aggregation plaintext. The combination of coarse and fine-grained aggregation results provides support of diverse data for effective power dispatching management.

TA: TA is primarily responsible for generating and managing all public parameters and secret keys for entities in the system. Meanwhile, it creates a Bloom filter for smart meters of each subarea by collecting a user's pseudonym. This Bloom filter will be sent to the corresponding users. The same operation is adaptable to the fog layer.

3.2. Adversary model

In the smart grid scenario, in order to pry into a user's private affairs, an eavesdropper may exist that can eavesdrop on the communication links between smart meters and fog nodes. At the same time, the active attacker may tamper with the transmission information and launch replay attacks to threaten the security of the smart grid. In our adversary model, we divide threats that may occur in the network into internal and external attacks.

Internal attack: The first category of internal attacks is composed of the malicious node attacks, which occur during the generation of the blockchain in the user and fog computing layers. For example, in the generation process of the blockchain, a malicious node pretends to be a legal node in the network, and initiates some active attacks (e.g., tampering, forgery, replay) to impair the authenticity and integrity of the user's private data. Therefore, the system should have the capability of identifying the legality of node identities in the consensus process. The second category of internal attacks is described as honest-but-curious in terms of fog and cloud nodes. For example, the fog node may be affected by undetected malware, and malware will eavesdrop on the data from devices, so we must ensure that the fog node does not observe the user's private data throughout the process. Similarly, the system should guarantee that the user's personal private data cannot be derived from the cloud server.

External attack: The attacker can eavesdrop and tamper with the transmitted data over communication links; it also can launch a replay attack. Therefore, the system must ensure that the attacker cannot successfully obtain the privacy information over the communication links and that it is immune to active attacks.

4. Double-blockchain assisted secure and anonymous data aggregation

In this section, we develop a DA-SADA scheme for the fog-enabled smart grid by integrating the blockchain, the Paillier cryptosystem, batch aggregation verification, and an anonymous authentication mechanism. It consists of four parts: system initialization, UA-blockchain generation, FA-blockchain generation, and service supporting.

4.1. System initialization

In our network scenario, the trusted third party TA is responsible for the system initialization, where there are three procedures that need to be executed in this system initialization process, that is, the generation of system parameters, the distribution of system parameters, and the generation of the Bloom filter.

The generation of system parameters: In the generation stage of system parameters, the TA selects the system security parameter κ to calculate two safe large primes $|p| = |q| = \kappa$. Consequently, it calculates $N = pq$ as the public key of the homomorphic encryption algorithm and $\lambda = \mathrm{lcm}(p-1,\ q-1)$ as the corresponding private key. Meanwhile, the system randomly selects $r \in Z_N^*$ and calculates $s = r^N \bmod N^2$. Let $g = N + 1$ and define the function as

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

Furthermore, for the sake of providing identity anonymity, the SM_{ij} chooses a random prime number X_{ij} to calculate its secret key $Y_{ij} = X_{ij}^{-1} \bmod N^2$; this public key X_{ij} is used to calculate the smart meter's pseudonym, that is, $\mathrm{Pseu}_{ij} = X_{ij} \bmod N^2$. Similarly, the fog node fog_j chooses a random prime number X_j as its public key and calculates $Y_j = X_j^{-1} \bmod N^2$ as its secret key to denote the fog device's pseudonym $\mathrm{Pseu}_j = X_j \bmod N^2$. Finally, the TA chooses the secure cryptographic hash function $H{:}\{0,1\}^* \to Z_N^*$.

The distribution of system parameters: With the generation of all system parameters $(\lambda, N, s, H, X_{ij}, X_j, Y_{ij}, Y_j)$, the public parameters (N, H) will be published online and the remainder of them will be allocated to the corresponding real entities. Specifically, keys $(X_{ij},\ Y_{ij},\ s)$, (X_j, Y_j), and λ are assigned, respectively, to the SM_{ij}, fog node fog_j and cloud server through the secret channel.

The generation of the Bloom filter: The TA collects the pseudonyms of smart meters to create a Bloom filter for each subarea. Similarly, the TA also collects the fog devices' pseudonym to create a Bloom filter in the fog layer. Specifically, in the user layer, the TA sets an array with θ bits; then, it uses a hash function to calculate the hash value of all pseudonyms in the same area. The element value of this array is set to one when the index value is equal to $H(\mathrm{Pseu}_{ij}) \bmod \theta$. Finally, the TA sends the Bloom filter to smart meters in the corresponding area. A similar operation will be implemented to generate the Bloom filter for the fog layer.

4.2. Generation of UA-blockchain

By considering the privacy leaks from the data analysis of the power consumption and tampering threat, the sensing device (i.e., smart meter) needs to encrypt the power consumption data of the user, and the relevant information needs to be digitally signed for integrity. This process is called transaction generation. Subsequently, the aggregation node aggregates the encrypted data and records the corresponding information into a block. Finally, the aggregation node generates the UA-blockchain by the consensus mechanism. The specific generation process of the UA-blockchain is shown in Fig. 3 and is represented below.

4.2.1. Transaction generation

The generation of power consumption ciphertext: For a subarea with n smart meters, in a certain time slot t_s, we denote the data item of the SM_{ij} as d_{ij}; then, each smart meter calculates ciphertext C_{ij} by the following formula:

$$\begin{aligned} C_{ij} &= g^{d_{ij}} \cdot r^N \bmod N^2 \\ &= (N+1)^{d_{ij}} \cdot r^N \bmod N^2 l \\ &= (1 + d_{ij}N) \cdot s \end{aligned} \tag{2}$$

where $1 \le i \le n,\ 1 \le j \le m$. We calculate $g = N + 1$ and obtain another form of the Paillier encryption algorithm $c = (1 + mN)r^N \bmod N^2$ according to the nature of $(1+N)^m \equiv (1 + mN) \bmod N^2$,

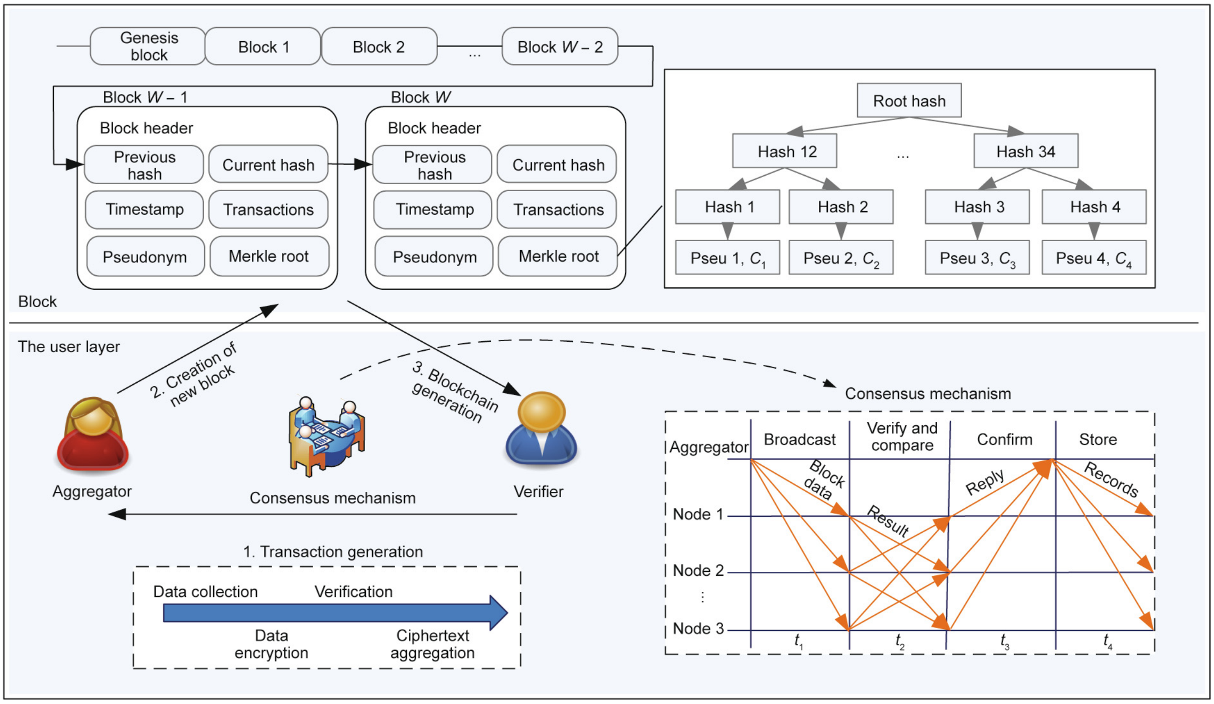

Fig. 3. Generation of UA-blockchain. This process includes three steps: transaction generation, creation of new block, and blockchain generation.

which is mainly used to avoid the cumbersome calculation in the encryption and decryption operation, thereby reducing the computational overhead.

The generation of the signature: It can obtain the signature σ_{ij}.

$$u_{ij} = H(C_{ij}||t_s) \tag{3}$$

$$\sigma_{ij} = H(u_{ij}||\text{Pseu}_{ij})^{Y_{ij}} \tag{4}$$

Next, the smart meter sends the report $(\sigma_{ij}||C_{ij}||t_s||\text{Pseu}_{ij})$ to the corresponding aggregation node in the user layer, where we choose the smart meter with the largest remaining computing resource as the aggregation node at the user layer.

Verification and ciphertext aggregation: After the aggregation node receives reports from each smart meter, it first checks the effectiveness of the user's pseudonym with the Bloom filter. Then, the timestamp is checked to confirm the validity of these reports. Finally, it uses the batch verification to verify signatures, and the specific expression is given as

$$\prod_{i=1}^{n} \sigma_{ij}^{X_{ij}} = \prod_{i=1}^{n} H\left[H(C_{ij}||t_s)||\text{Pseu}_{ij}\right] \bmod N^2 \tag{5}$$

where this equation is derived from the aggregation operation and the concrete values of the public and private keys. The detailed expression is written as

$$\begin{aligned}\prod_{i=1}^{n} \sigma_{ij}^{X_{ij}} &= \prod_{i=1}^{n} H(u_{ij}||\text{Pseu}_{ij})^{Y_{ij}X_{ij}} \bmod N^2 \\ &= \prod_{i=1}^{n} H\left[H(C_{ij}||t_s)||\text{Pseu}_{ij}\right]^{Y_{ij}X_{ij}} \bmod N^2 \\ &= \prod_{i=1}^{n} H\left[H(C_{ij}||t_s)||\text{Pseu}_{ij}\right] \bmod N^2\end{aligned} \tag{6}$$

After the successful verification of the smart meters' signatures, the aggregation node performs an aggregation operation to obtain the aggregated ciphertext C_j for subarea j:

$$C_j = \prod_{i=1}^{n} C_{ij} \bmod N^2 \tag{7}$$

Consequently, the aggregated ciphertext C_j is combined with other related information to generate the transaction $T_x = (C_j, \text{Pseu}_{ij}, t_s)$.

4.2.2. Creation of the new block

The aggregation node records the transaction $T_x = (C_j, \text{Pseu}_{ij}, t_s)$ in a block and broadcasts this block in the subarea j for information authentication. This block still includes three other elements, that is, the Merkle root and previous and current hashes. The value of the Merkle root is achieved by hashing the ciphertext C_j and the pseudonym in the Merkle tree, as shown in Fig. 3. The hash value of the current block is calculated as following equation:

$$\begin{aligned}H_{\text{curr-block}} = \text{SHA256}\Bigg(&\text{index} + H_{\text{prev-block}} + \text{Pseu}_{ij} + \text{timestamp} \\ &+ C_j + \sum_{ij} \text{transactions}_{ij}\Bigg)\end{aligned} \tag{8}$$

where this calculation process implies that once the block is added into a chain, it is difficult to tamper with the block content since the hash value of the previous block is involved in calculating the hash value of the current block.

4.2.3. Blockchain generation

After the aggregation node creates a new block, the new block is broadcast in this subarea. The ordinary node in this subarea verifies records in this new block, and each node only verifies the data related to itself for meeting the real-time scheduling requirement in the smart grid. If it is consistent with the original data, it passes the verification and broadcasts the verification result to other nodes in the user layer. After collecting the correctness confirmation message sent by the other $2n/3+1$ nodes or more, this new block is considered to be valid and added to the UA-blockchain. In our blockchain network, we assume that the number of malicious nodes should be less than 1/3 of the total number of network nodes. Because we define that a new block can only be added to the blockchain when it passes the verifications of $2n/3+1$ nodes or more, we set such threshold value for security consideration. It also implies the attacker can tamper the information in the block successfully only when it captures more than 2/3 nodes of the network. The specific consensus process is shown in Fig. 3.

4.3. Generation of FA-blockchain

Similarly, in the fog computing layer, the generation of the FA-blockchain consists of the transaction generation, new block creation, and blockchain generation.

4.3.1. Transaction generation

The transaction generation of the fog layer is similar to that in the user layer. First, when the fog node j receives encrypted data from the UA-blockchain, these encrypted data will be digitally signed for integrity at fog node j. Then, the selected aggregation node at the fog layer performs the aggregation operation for all of the $C_j, j \in \{1, 2, \ldots, m\}$, that is, it obtains the secondary aggregation result. Similarly, we choose the fog node with the largest remaining computing resource as the aggregation node.

The generation of the signature: When the jth fog node fog$_j$ receives the aggregated power consumption ciphertext C_j of the corresponding subarea, it can calculate the signature σ_j:

$$u_j = H(C_j||t_s) \tag{9}$$

$$\sigma_j = H(u_j||\text{Pseu}_j)^{Y_j} \tag{10}$$

Next, this fog node sends the report $(\sigma_j||C_j||t_s||\text{Pseu}_j)$ to the corresponding aggregation node in the fog layer.

Verification and ciphertext aggregation: After the aggregation node receives the reports from each fog computing device, it first checks the effectiveness of the fog device's pseudonym with the Bloom filter. Then, the timestamp is checked to confirm the validity of these reports. Finally, it uses the batch verification to verify these signatures, and the specific expression is given as

$$\prod_{j=1}^{m} \sigma_j^{X_j} = \prod_{j=1}^{m} H(u_j||\text{Pseu}_j) \bmod N^2 \tag{11}$$

where this equation is derived from the aggregation operation and the concrete values of the public and private keys. The detailed expression is written as

$$\begin{aligned}\prod_{j=1}^{m} \sigma_j^{X_j} &= \prod_{j=1}^{m} H\left[H(C_j||t_s)||\text{Pseu}_j\right]^{Y_jX_j} \bmod N^2 \\ &= \prod_{j=1}^{m} H\left[u_j||\text{Pseu}_j\right]^{Y_jX_j} \bmod N^2 \\ &= \prod_{j=1}^{m} H\left[u_j||\text{Pseu}_j\right] \bmod N^2\end{aligned} \tag{12}$$

After the successful verification of the smart meters' signatures, the aggregation node performs an aggregation operation to obtain the secondary aggregation ciphertext C_{AS} for all subareas.

$$C_{\mathrm{AS}} = \prod_{j=1}^{m} C_j \bmod N^2 \tag{13}$$

Consequently, the aggregated ciphertext is combined with other related information to generate the transaction $T'_x = (C_{\mathrm{AS}}, \mathrm{Pseu}_j, t_s)$.

4.3.2. Creation of the new block

The aggregation node at the fog layer records the transaction $T'_x = (C_{\mathrm{AS}}, \mathrm{Pseu}_j, t_s)$ in a block and broadcasts this block to other fogs for information authentication. Similar to the creation of the block in the user layer, this block includes the transaction, the Merkle root and previous and current hashes. The hash value of the current block is calculated as the following equation.

$$\begin{aligned} H'_{\text{curr-block}} = \mathrm{SHA256}\Bigg(& \text{index} + H_{\text{prev-block}} + \mathrm{Pseu}_j + \text{timestamp} \\ & + C_{\mathrm{AS}} + \sum_j \text{transactions}_j \Bigg) \end{aligned} \tag{14}$$

4.3.3. Blockchain generation

After the aggregation node creates a new block in the fog computing layer, the new block is broadcast to other fog nodes and added into the FA-blockchain through the consensus mechanism. The consensus mechanism is similar to that of the user layer. First, the ordinary node in the fog computing layer verifies the records in this new block and each node only verifies the data related to itself. If it is consistent with the original data, it passes the verification and broadcasts the verification result to other nodes in the fog computing layer. After collecting the correctness confirmation message sent by the other $2m/3 + 1$ fog nodes or more, this block is considered to be valid and added to the FA-blockchain.

4.4. Service supporting

When the cloud server receives the FA-blockchain from the fog computing layer, it reads the secondary aggregation ciphertext and decrypts the ciphertext by using the Paillier decryption algorithm. To leverage the Paillier decryption algorithm effectively, we further specify the components of Eq. (13), that is, Eq. (13) can be rewritten as

$$\begin{aligned} C_{\mathrm{AS}} &= \prod_{j=1}^{m} C_j \bmod N^2 \\ &= \prod_{j=1}^{m}\left(\prod_{i=1}^{n} C_{ij} \bmod N^2\right) \\ &= \prod_{j=1}^{m}\left(\prod_{i=1}^{n} g^{d_{ij}} \cdot r_j^N \bmod N^2\right) \\ &= \prod_{i=1}^{n}\left(\prod_{j=1}^{m} g^{d_{ij}} \cdot r_j^N \bmod N^2\right) \\ &= \prod_{i=1}^{n}\left(g^{d_{i1}} \cdot g^{d_{i2}} \cdot \dots \cdot g^{d_{im}} \bmod N^2\right)\left(\prod_{j=1}^{m} r_j\right)^N \bmod N^2 \\ &= g^{a_1 \sum_{i=1}^{n} d_{i1}} \cdot g^{a_2 \sum_{i=1}^{n} d_{i2}} \cdot \dots \cdot g^{a_m \sum_{i=1}^{n} d_{im}} \left(\prod_{j=1}^{m} r_j\right)^N \bmod N^2 \\ &= g^{a_1 \sum_{i=1}^{n} d_{i1} + a_2 \sum_{i=1}^{n} d_{i2} + \dots + a_m \sum_{i=1}^{n} d_{im}} \left(\prod_{j=1}^{m} r_j\right)^N \bmod N^2 \end{aligned} \tag{15}$$

Meanwhile, we respectively define symbols M and R as

$$M = a_1 \sum_{i=1}^{n} d_{i1} + a_2 \sum_{i=1}^{n} d_{i2} + \dots + a_m \sum_{i=1}^{n} d_{im} \tag{16}$$

$$R = \prod_{j=1}^{m} r_j \tag{17}$$

Accordingly, the cloud server can organize the ciphertext C_{AS} into the following format, which is in accordance with the ciphertext form of Paillier encryption.

$$C = g^M \cdot R^N \bmod N^2 \tag{18}$$

Consequently, the cloud server can use the Paillier decryption algorithm to decrypt the aggregated ciphertext directly and obtain the aggregated plaintext M.

$$\begin{aligned} M &= \frac{L\left(C^{\lambda} \bmod N^2\right)}{L\left(g^{\lambda} \bmod N^2\right)} \\ &= \frac{L\left(C^{\lambda} \bmod N^2\right)}{L\left[(1+N)^{\lambda} \bmod N^2\right]} \\ &= L\left(C^{\lambda} \bmod N^2\right) \cdot \lambda^{-1} \end{aligned} \tag{19}$$

Finally, Horner's rule is employed to complete the high-speed analysis of the aggregated plaintext and obtain fine-grained aggregation results; the detailed solving process is shown as Algorithm 1. In this algorithm, the coefficient denotes the total power consumption of subarea j, which is defined as

$$\mathrm{UA}_j = \sum_{i=1}^{n} d_{ij} \tag{20}$$

Due to the values of these coefficients, it achieves the fine-grained aggregation successfully, that is, it not only obtains the entire power consumption of the network but also recovers the subarea's data.

Algorithm 1. Horner rule-based analytical algorithm.

```
Input:
  $M$ and $R$.
Output:
  Total power consumption $\mathrm{UA}_j$ in each subarea $j$,
  $j = 1, 2, \dots, m$.
Begin
    $x_0 \leftarrow M/R$, $a_1 = R^1, a_2 = R^2, \dots, a_m = R^m$;
  $x_0 = \mathrm{UA}_1 + R^1\mathrm{UA}_2 + \dots + R^{m-1}\mathrm{UA}_m$;
    For $j \leftarrow 1$ to $m$ do
      $\mathrm{UA}_j \leftarrow x_{j-1} \bmod R$;
      $x_j \leftarrow x_{j-1} \bmod R$;
    End for
    Obtain $(\mathrm{UA}_1, \mathrm{UA}_2, \dots, \mathrm{UA}_m)$.
End
```

Once the cloud server gains the power consumption of each subarea through the above operations, these fine-grained data can be explored to predict the power usage trend of each subarea, and then provide decision support for power dispatching and price adjustment. Accordingly, the smart contract enables these decisions to be executed automatically and develops the time-of-use pricing feedback strategy to encourage users to adjust their electricity use habits for alleviating the burden of the power grid and improving the power utilization efficiency.

With the accumulation of data, the blockchain sharing ledger will become increasingly larger, which is called blockchain bloat. For example, in the past nine years, the size of the Bitcoin system ledger has reached 153.1 GB [32]. All historical transaction items of Bitcoin need to be kept for a long time because they are used to calculate account balances. For the proposed aggregation mechanism in this paper, the smart meter's data item of the new generation does not rely on the previous one, thus there is no need to save all the data items on each node. We recommend regularly cleaning out obsolete data items and releasing storage space in the relevant nodes.

5. Security and performance evaluations

In this section, we will discuss the security and anonymity properties of the proposed scheme, and analyze the performance in terms of the computation cost. In particular, we perform a quantitative analysis on the successful probability of tampering attacks under different scenarios, which proves the high security of our proposed scheme. Furthermore, the computation costs of the identity authentication and whole system are given in detail, and they show that the proposed scheme is lightweight and more suitable for systems with real-time requirements.

5.1. Security analysis

Data confidentiality: In our defined threat model, the transmission of users' power consumption data over channels is subjected to the eavesdropping attack, and the fog and cloud nodes are both honest-but-curious. To guarantee the confidentiality of users' privacy data, Paillier encryption is employed to encrypt these consumption data in the form of ciphertext $C_{ij} = g^{d_{ij}} \cdot r^N \bmod N^2$. Even if the eavesdroppers observe all of these data and know the encryption algorithm, they cannot decrypt the ciphertext to reveal users' data without the private key because an encrypted result of Paillier encryption is semantically secure from the chosen plaintext attack [33]. Similarly, the aggregation execution objects of fog and cloud nodes are all encrypted data, so the fog node cannot obtain the user's real data without the corresponding private key. Although the cloud server can recover the aggregated plaintext of each subarea as $\mathrm{UA}_j = \sum_{i=1}^{n} d_{ij}$ by using the additive homomorphism of the Paillier algorithm, it still cannot deduce the original meter data of each user. Therefore, our developed scheme provides strong confidentiality for users' power consumption data, that is, it protects the users' privacy information effectively.

Data integrality and validity: Aiming at resisting the active attacks (e.g., tampering, forgery, replay) on the smart meter's data, in our scheme, the user signs the ciphertext C_{ij} and timestamp t_s as $\sigma_{ij} = H\left[H(C_{ij}||t_s)||\mathrm{Pseu}_{ij}\right]^{Y_{ij}}$ by using the batch aggregation signature before sending it to the upper layer. Only when the $\prod_{i=1}^{n} \sigma_{ij}^{X_{ij}}$ is equal to $\prod_{i=1}^{n} H\left[H(C_{ij}||t_s)||\mathrm{Pseu}_{ij}\right] \bmod N^2$, can the receiver confirm that the received information has not been tampered with. Obviously, once the data $(\sigma_{ij}||C_{ij}||t_s||\mathrm{Pseu}_{ij})$ is tampered with, this equation will not be established. That is, even if the attacker successfully modifies the information or launches replay attacks, the receiver can detect these threats effectively. As a result, our developed scheme guarantees the integrity and validity of the private data. It also provides the same security protections in the fog layer.

Identity anonymity and authenticity: The user identity is usually associated with the private information, and the disclosure of the user identity information can often cause a series of hazards. In the proposed scheme in this paper, the identities of smart meters and fog devices always exist in a pseudonym form, that is, $\mathrm{Pseu}_{ij} = X_{ij} \bmod N^2$ and $\mathrm{Pseu}_j = X_j \bmod N^2$, respectively, where the public keys X_{ij} and X_j are randomly selected by the user and fog device, respectively, and the generated pseudonyms Pseu_{ij} and Pseu_j are random and are not associated with the true identity of the user and fog device. Even if the malicious attacker decrypts the meter's data of users successfully, it still means nothing because it cannot obtain the real identity of the user. Thus, our scheme realizes the anonymity of the user identity. At the same time, an illegal node may exist that attempts to impersonate the legal user's identity; however, our identity authenticity mechanism can identify this identity fraud behavior since we have already collected the legal pseudonym in advance and mapped it in the Bloom filter. It can quickly determine whether the node's pseudonym is in the Bloom filter by the querying operation.

5.2. Successful attacking probabilities

According to the threat model definition in Section 3, we choose two typical attacks to evaluate their impacts on aggregation results, that is, tampering attacks in nodes and over links. To demonstrate the advantages of our proposed solution, we comparatively analyze the successful probability of tampering attacks under different solutions.

5.2.1. Tampering attack in nodes

In our threat model, we assume that the total number of smart meters that attackers need to manipulate is w if they want to successfully launch a tampering attack, and the total number that attackers need to manipulate of fog nodes is f. To make it easier to understand, we suppose that the compromised probability of each smart meter is independent and denoted as α_i, where $i = 1, 2, \ \ldots, w, \ldots, nm$ and $0 \le \alpha_i \le 1$. Similarly, the compromised probabilities of the fog node and cloud server are represented, respectively, by β_j, $j = 1, 2, \ldots, f, \ldots, m$, $0 \le \beta_j \le 1$ and γ. Meanwhile, we assume the intercepted probability of the smart meter's secret key by a malicious node is independent and set to be ∂_i, where $i = 1, 2, \ldots, w, \ldots, nm$ and $0 \le \partial_i \le 1$.

Therefore, the successful probability of tampering attack under the traditional secure scheme can be given as

$$P_{\mathrm{node}} = \frac{1}{3}\left[\left(\prod_{i=1}^{w}\alpha_i\right)\left(\prod_{i=1}^{w}\partial_i\right) + \left(\prod_{j=1}^{f}\beta_j\right)\left(\prod_{i=1}^{fn}\partial_i\right) + \gamma\right] \tag{21}$$

where the weight is 1/3, indicating that the attacker chooses to attack three category nodes equally. This traditional secure scheme is that the data are transmitted among nodes without considering blockchain, but it has other same secure mechanism with our proposed scheme.

For the proposed scheme in this paper, the blockchain-based secure scheme can tolerate less than 1/3 of compromised nodes due to the existence of the consensus mechanism. Based on this conclusion, we define the threshold $\psi = \mathrm{ceil}[(2/3)(n-1)+1]$ for each subarea, where the function ceil() is defined to return the smallest integer that is greater than or equal to the specified expression. In the meantime, the other threshold $\psi' = \mathrm{ceil}[(2/3)(m-1)+1]$ is given for the fog layer. Consequently, the successful probability of a tampering attack under our proposed scheme can be written as

$$P'_{\mathrm{node}} = \frac{1}{3}\left[\left(\prod_{i=1}^{m\psi}\alpha_i\right)\left(\prod_{i=1}^{m\psi}\partial_i\right) + \left(\prod_{j=1}^{\psi'}\beta_j\right)\left(\prod_{i=1}^{n\psi'}\partial_i\right) + \gamma\right] \tag{22}$$

5.2.2. Tampering attack over links

In this part, we consider the attack that intercepts or forges data packets over the communication channels.

For the traditional secure scheme, the attacker may launch an attack to tamper with power data before the fog node or cloud server receives these data. This type of attack requires the successful intrusion into the communication channel and obtains the private key of the sender node to successfully modify the data. Therefore, we denote η_i, $i = 1, 2, \ldots, w, \ldots, nm$, $0 \le \eta_i \le 1$ as the successful probability of an intercepting attack over the communication link between smart meter and fog node, and denote $\overline{\eta}_j$, $j = 1, 2, \ldots, f, \ldots, m$, $0 \le \overline{\eta}_j \le 1$ as the successful probability of an intercepting attack over the communication link between the fog node and cloud server. The independent successful probability of this kind of attack is

$$P_{\text{link}} = \frac{1}{2}\left[\left(\prod_{i=1}^{w}\eta_i\right)\left(\prod_{i=1}^{w}\partial_i\right) + \left(\prod_{j=1}^{f}\overline{\eta}_j\right)\left(\prod_{i=1}^{fn}\partial_i\right)\right] \tag{23}$$

where the weight is 1/2, indicating that the attacker chooses to attack the two kinds of communication links equally.

For the proposed scheme in this paper, considering that the privacy data are encapsulated into the blockchain between the user and fog layer, as well as the link between the fog layer, and cloud server, we know that generally, the data in the blockchain are not tamperable, so we do not consider the successful possibility of a tampering attack in the communication links among the user layer, fog layer, and cloud server. However, a consensus process needs to be executed before the blockchain is formed, in which nodes of the user layer need to communicate with each other to form an internal communication network, and in this communication network, it will be confronted with the tampering attack. This threat also exists in the fog layer. Therefore, we will analyze the successful probability of a tampering attack over these two internal networks. First, we assume that there are $x_{\text{uc}} = m \cdot \text{ceil}[\psi(\psi - 1)/2]$ communication channels in the user layer, and the successful probability of intruding a communication channel at the user layer is denoted as $\eta_{x_{\text{uc}}}$, where $x_{\text{uc}} = 1, 2, \ldots, l, \ldots, L$ and $0 \le \eta_{x_{\text{uc}}} \le 1$. Meanwhile, we assume that there are $x_{\text{fc}} = \text{ceil}[\psi'(\psi' - 1)/2]$ communication channels in the fog computing layer, and the successful probability of intruding a communication channel at the fog layer is denoted as $\overline{\eta}_{x_{\text{fc}}}$, where $x_{\text{fc}} = 1, 2, \ldots, \mathscr{K}, \ldots, K$ and $0 \le \overline{\eta}_{x_{\text{fc}}} \le 1$. Therefore, the independent successful probability for tampering with data in the proposed hierarchical blockchain network is

$$P'_{\text{link}} = \frac{1}{2}\left[\left(\prod\nolimits_{x_{\text{uc}}=1}^{l}\eta_{x_{\text{uc}}}\right)\left(\prod\nolimits_{i=1}^{m\psi}\partial_i\right) + \left(\prod\nolimits_{x_{\text{fc}}=1}^{\mathscr{K}}\overline{\eta}_{x_{\text{fc}}}\right)\left(\prod\nolimits_{i=1}^{n\psi'}\partial_i\right)\right] \tag{24}$$

Finally, we assume that the tampering attack in nodes and over links is independent, and the probability of being chosen by the attacker is equal. Consequently, the total successful probability of tampering attacks for the traditional and our proposed schemes are, respectively, given as

$$P_{\text{tradition}} = \frac{1}{2}(P_{\text{node}} + P_{\text{link}}) \tag{25}$$

$$P_{\text{proposed}} = \frac{1}{2}(P'_{\text{node}} + P'_{\text{link}}) \tag{26}$$

where the weight is 1/2, indicating that the successful probability of an attacker launching two kinds of attacks is independent and equal.

5.2.3. Successful probabilities

In the previous two parts, we analyzed the successful probability of the tampering attack for the traditional and our proposed schemes from a theoretical perspective. To show the analysis results more intuitively, we use the Monte Carlo simulation method to further analyze the successful probability. In this simulation scenario, we assume that there are 20 smart meters in each subarea and 1 cloud server in the service supporting layer, and the number of fog nodes is 50. Then, we assume that the probability that attackers need to manipulate smart meters is 10% to 100%; thus, the w is variable from 100 to 1000 in the entire network. Meanwhile, we define that the range of variables $\alpha, \beta, \partial, \eta$ and $\overline{\eta}$ all vary from 0.9 to 1, and the range of γ is set to be [0, 0.1]. The values of variables $\alpha, \beta, \partial, \eta$ and $\overline{\eta}$ are randomly selected within their ranges, and we execute the experiment 1000 times to evaluate the average value of the simulation results. The experiment runs on a notebook with an Intel Core i5-7200U CPU @ 2.50 GHZ, with 8.00 GB RAM.

Fig. 4 depicts the interrelation between the successful attacking probability and the total number of smart meters that attackers need to manipulate. Notably, the successful probability exhibits a continuous decline with the increase of the number of the manipulated smart meters, and our proposed scheme demonstrates a significant advantage in the reduction of security threats. In particular, the successful attacking probability approaches 0 in our scheme when the total number that attackers need to manipulate is more than 500. The main reason for this result is that our proposed scheme designs two consensus mechanisms in the generation of the UA-blockchain and FA-blockchain, and the consensus mechanism needs group verification. Therefore, the use of the double-blockchain significantly enhances the robustness of the system.

5.3. Computation cost

In this subsection, we analyze the computation costs of identity authentication and the entire system. In the simulation scenario, we assume that the number of fog nodes is variable from 5 to 50. Meanwhile, we set the error probability of the Bloom filter to 0.01, and define the RSA modulus N and parameter p as 1024 bits and 160 bits, respectively. Although the content-based Bloom filter usually has conflicts, the conflict probability is very small. For example, in the case of using seven different hash functions, to use a bit string of 2 MB size, the overall error rate is less than 0.01. Therefore, it is reasonable to set the error probability of the Bloom filter to 0.01. For convenience of explanation, we denote $T_{\text{E1}}, T_{\text{E2}}, T_{\text{M}}$ and 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 pairing-based cryptography (PBC) library to implement these operations. The

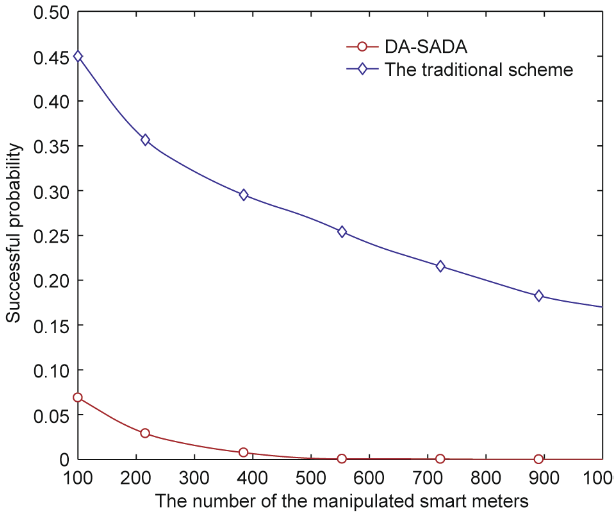

Fig. 4. Successful attacking probabilities under different solutions.

Table 1
Operation notations and time costs.

Notation	Description	Time cost (ms)
T_{E1}	Exponentiation operation in $Z_{N^2}^*$	1.60
T_{E2}	Exponentiation operation in G	1.62
T_{M}	Multiplication operation	0.06
T_{P}	Pairing operation	17.70

data set of simulation is from Commission for Energy Regulation Ireland [34]. Table 1 lists the operation notations and their time costs in the evaluation process.

Fig. 5 shows the time cost of identity authentication with and without the Bloom filter. Observing from this figure, we can find that the time cost of the traditional scheme without the Bloom filter grows sharply with the increase of the number of smart meters, but our proposed scheme has a limited increasing range and the time cost is much lower than the traditional scheme. This is because the Bloom filter uses multiple hash functions to improve space utilization, which greatly improves the query efficiency of the authentication process.

Subsequently, for the sake of comprehensively displaying the computation cost, we analyze the computation cost of the entire system with our developed scheme, and conduct a comparison with two benchmark schemes, that is, the security-enhanced data aggregation scheme (SEDA) [13] and a lightweight privacy-preserving data aggregation scheme for edge computing (LPDA-EC) [21]. Because the computation cost of the hash operation is negligible compared with exponentiation and multiplication operations, we do not consider the cost of the hash operation in our evaluation process.

Specifically, in the user layer, the generation of ciphertext $C_{ij} = (1 + d_{ij}N) \cdot s$ and signature $\sigma_{ij} = H(u_{ij}||\mathrm{Pseu}_{ij})^{Y_{ij}}$ requires two multiplication operations T_{M} and one exponentiation operation T_{E1} in $Z_{N^2}^*$, respectively. After the aggregation node in the user layer receives the report $(\sigma_{ij}||C_{ij}||t_s||\mathrm{Pseu}_{ij})$ from n smart meters, it first authenticates the validity and integrity of the received data by batch verification, which includes n multiplication operations T_{M} and one exponentiation operation T_{E1} in $Z_{N^2}^*$. Next, the aggregation computation of the user's data needs n multiplication operations T_{M}. Finally, the aggregation node sends the report to the fog layer. In the fog layer, to generate the signature σ_j, one exponentiation operation T_{E1} in $Z_{N^2}^*$ is needed; then, fog nodes send the report $(\sigma_j||C_j||t_s||\mathrm{Pseu}_j)$ to the aggregation node at the fog layer, and the aggregation node first authenticates the received data by batch verification, which includes m multiplication operations T_{M} and one exponentiation operation T_{E1} in $Z_{N^2}^*$. After the successful authentication, the aggregation node aggregates the first-level aggregation report $C_j, j = 1, 2, \ldots, m$, and the aggregation computation needs m multiplication operations T_{M}. Then, the fog node sends the aggregation report to the upper layer. Upon receiving the report from the fog node, the cloud server decrypts the aggregation report, and this Paillier decryption includes one exponentiation operation T_{E1} and one multiplication operation T_{M}. From the above analysis, the entire calculation process of our proposed scheme includes $(4mn + 2m + 1)T_{\mathrm{M}} + (mn + 2m + 2)T_{\mathrm{E1}}$ operations. Similarly, we can obtain the computation costs of other schemes. The specific operation statistics of these schemes are shown in the Table 2.

In Fig. 6, similar to the computation cost of identity authentication, the total computation cost of the system is proportional to the number of smart meters. Meanwhile, we can observe that our proposed scheme achieves a significant reduction in the total computation cost compared with SEDA and LPDA-EC. For example, when the number of smart meters is 500, the total computation cost of our proposed scheme is 10^3 ms, which reduces by 80% and 60% that of SEDA and LPDA-EC, respectively. Furthermore, the reduction of the computation cost will become more pronounced with the increase of the number of smart meters. This is mainly because the required time for bilinear pairing is much larger than that of other operations, and both SEDA and LPDA-EC include the expensive bilinear pairing operation during the verification process. However, in our proposed scheme, the use of the pairing calculation is effectively avoided, which significantly reduces the computation cost at the same time.

From the above security and performance analysis results, we can conclude that the proposed security and anonymous data aggregation scheme significantly reduces the system computation

Table 2
Time costs.

Scheme	Operation
DA-SADA	$(4mn + 2m + 1)T_{\mathrm{M}} + (mn + 2m + 2)T_{\mathrm{E1}}$
SEDA	$2T_{\mathrm{P}} + 4T_{\mathrm{E1}} + (6nm + 3)T_{\mathrm{E2}} + (2nm + 1)T_{\mathrm{M}}$
LPDA-EC	$2T_{\mathrm{P}} + 4T_{\mathrm{E1}} + (3nm + 3)T_{\mathrm{E2}} + (2nm + 1)T_{\mathrm{M}}$

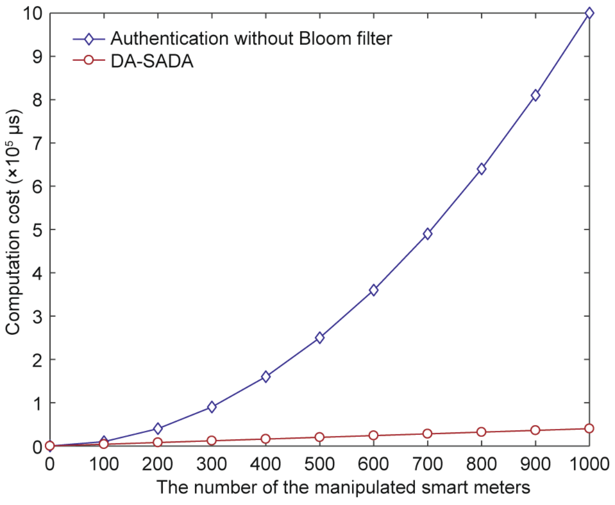

Fig. 5. Time cost of identity authentication.

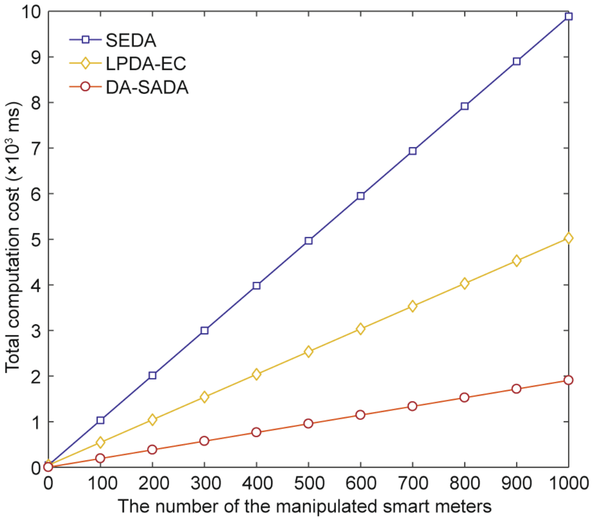

Fig. 6. Total computation cost of the system.

cost while providing strong security and anonymity protections. Moreover, it is more suitable for systems with real-time high-frequency data collection and aggregation requirements in the smart grid.

6. Conclusions

The smart grid can achieve reliable and stable services by collecting and analyzing the users' electricity consumption data, but the users' security and privacy are usually threatened during these operations. Therefore, we propose a DA-SADA scheme. Specifically, we construct a security-enhanced three-tier architecture by combining fog computing and the blockchain, and the local resources are exploited effectively. Subsequently, a lightweight secure aggregation mechanism is developed to ensure the confidentiality, integrity, and authenticity of private data. In particular, in order to realize the flexible regulation of power, we design the double-blockchain to achieve fine-grained aggregation of the users' power consumption data, and the double-consensus in the formation of the double-blockchain further enhances the security of the system. Finally, the security analysis confirms the high security of our proposed scheme, and the comparison analysis of computation costs in the entire system further validates its performance advantage, providing a more suitable solution for systems with real-time requirements. Although our proposed scheme provides an efficient and secure data collection mechanism for smart grid, it still lacks an efficient and smart method to select aggregation node. Therefore, in future work, we plan to develop a dynamic and smart aggregation node selection mechanism to improve the applicableness of developed scheme in the real network scenario by integrating machine learning method.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (61971235, 61871412, and 61771258), the Six Talented Eminence Foundation of Jiangsu Province (XYDXXJS-044), the China Postdoctoral Science Foundation (2018M630590), the 333 High-level Talents Training Project of Jiangsu Province, the 1311 Talents Plan of Nanjing University of Posts and Telecommunications (NUPT), the Open Research Fund of Jiangsu Engineering Research Center of Communication and Network Technology, NUPT (JSGCZX17011), the Scientific Research Foundation of NUPT (NY218058), and the Open Research Fund of Anhui Provincial Key Laboratory of Network and Information Security (AHNIS2020001).

Compliance with ethics guidelines

Siguang Chen, Li Yang, Chuanxin Zhao, Vijayakumar Varadarajan, and Kun Wang declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Ketter W, Collins J, Saar-Tsechansky M, Marom O. Information systems for a smart electricity grid: emerging challenges and opportunities. ACM Trans Manage Inf Syst 2018;9(3):1–22.
[2] Chen S, Wen H, Wu J, Lei W, Hou W, Liu W, et al. Internet of Things based smart grids supported by intelligent edge computing. IEEE Access 2019;7(1):74089–102.
[3] Asghar MR, Dan G, Miorandi D, Chlamtac I. Smart meter data privacy: a survey. IEEE Commun Surv Tutorials 2017;19(4):2820–35.
[4] Wang Y, Chen Q, Hong T, Kang C. Review of smart meter data analytics: applications, methodologies, and challenges. IEEE Trans Smart Grid 2019;10(3):3125–48.
[5] Abdallah A, Shen XS. A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid. IEEE Trans Smart Grid 2018;9(1):396–405.
[6] Chen S, Wang K, Zhao C, Zhang H, Sun Y. Accelerated distributed optimization design for reconstruction of big sensory data. IEEE Internet Things J 2017;4(5):1716–25.
[7] Chen S, Wang Z, Zhang H, Yang G, Wang K. Fog-based optimized Kronecker-supported compression design for industrial IoT. IEEE Trans Sustain Comput 2020;5(1):95–106.
[8] Yan Y, Qian Y, Sharif H, Tipper D. A survey on cyber security for smart grid communications. IEEE Commun Surv Tutorials 2012;14(4):998–1010.
[9] Chen Y, Martinez-Ortega JF, Castillejo P, Lopez L. A homomorphic-based multiple data aggregation scheme for smart grid. IEEE Sens J 2019;19(10):3921–9.
[10] Gai K, Wu Y, Zhu L, Qiu M, Shen M. Privacy-preserving energy trading using consortium blockchain in smart grid. IEEE Trans Ind Inf 2019;15(6):3548–58.
[11] Liang G, Weller SR, Zhao J, Luo F, Dong ZY. The 2015 ukraine blackout: implications for false data injection attacks. IEEE Trans Power Syst 2017;32(4):3317–8.
[12] Lu R, Liang X, Li X, Lin X, Shen X. EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications. IEEE Trans Parallel Distrib Syst 2012;23(9):1621–31.
[13] Ni J, Alharbi K, Lin X, Shen X. Security-enhanced data aggregation against malicious gateways in smart grid. In: Proceedings of the 2015 IEEE Global Communications Conference; 2015 Dec 6–10; San Diego, CA, USA; 2015.
[14] Gope P, Sikdar B. An efficient data aggregation scheme for privacy-friendly dynamic pricing-based billing and demand-response management in smart grids. IEEE Internet Things J 2018;5(4):3126–35.
[15] Liu Y, Guo W, Fan C, Chang L, Cheng C. A practical privacy-preserving data aggregation (3PDA) scheme for smart grid. IEEE Trans Ind Inf 2019;15(3):1767–74.
[16] Li S, Xue K, Yang Q, Hong P. PPMA: privacy-preserving multisubset data aggregation in smart grid. IEEE Trans Ind Inf 2018;14(2):462–71.
[17] Peng L, Dhaini AR, Ho P. Toward integrated cloud-fog networks for efficient IoT provisioning: key challenges and solutions. Future Gener Comput Syst 2018;88:606–13.
[18] Lu R, Heung K, Lashkari AH, Ghorbani AA. A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT. IEEE Access 2017;5:3302–12.
[19] Huang C, Liu D, Ni J, Lu R, Shen X. Reliable and privacy-preserving selective data aggregation for fog-based IoT. In: Proceedings of the 2018 IEEE International Conference on Communications; 2018 May 20–24; Kansas City, MO, USA; 2018.
[20] Lyu L, Nandakumar K, Rubinstein B, Jin J, Bedo J, Palaniswami M. PPFA: privacy preserving fog-enabled aggregation in smart grid. IEEE Trans Ind Inf 2018;14(8):3733–44.
[21] Zhang J, Zhao Y, Wu J, Chen B. LPDA-EC: a lightweight privacy-preserving data aggregation scheme for edge computing. In: Proceedings of the 2018 IEEE 15th International Conference on Mobile Ad Hoc Sensor Systems; 2018 Oct 9–12; Chengdu, China; 2018.
[22] Zhu L, Li M, Zhang Z, Xu C, Zhang R, Du X, et al. Privacy-preserving authentication and data aggregation for fog-based smart grid. IEEE Commun Mag 2019;57(6):80–5.
[23] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system [Internet]. Bitcoin; 2018 [cited 2019 Aug 21]. Available from: https://bitcoin.org/bitcoin.pdf.
[24] Liang G, Weller SR, Luo F, Zhao J, Dong ZY. Distributed blockchain-based data protection framework for modern power systems against cyber attacks. IEEE Trans Smart Grid 2019;10(3):3162–73.
[25] Fan M, Zhang X. Consortium blockchain based data aggregation and regulation mechanism for smart grid. IEEE Access 2019;7:35929–40.
[26] Guan Z, Si G, Zhang X, Wu L, Guizani N, Du X, et al. Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities. IEEE Commun Mag 2018;56(7):82–8.
[27] Yang R, Yu FR, Si P, Yang Z, Zhang Y. Integrated blockchain and edge computing systems: a survey, some research issues and challenges. IEEE Commun Surv Tutorials 2019;21(2):1508–32.
[28] Xiong Z, Zhang Y, Niyato D, Wang P, Han Z. When mobile blockchain meets edge computing. IEEE Commun Mag 2018;56(8):33–9.
[29] Wood G. Ethereum: a secure decentralised generalised transaction ledger. Ethereum Proj Yellow Pap 2014;151:1–32.
[30] Hassan MU, Rehmani MH, Chen J. Privacy preservation in blockchain based IoT systems: integration issues, prospects, challenges, and future research directions. Future Gener Comput Syst 2019;97:512–29.
[31] Bose P, Guo H, Kranakis E, Maheshwari A, Morin P, Morrison J, et al. On the false-positive rate of Bloom filters. Inf Process Lett 2008;108(4):210–3.
[32] Bitcoin.com. Blockchain size [Internet]. Saint Bitts LLC; 2018 [cited 2019 Aug 21]. Available from: https://charts.bitcoin.com/chart/blockchainsize.
[33] Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of the 1999 Annual International Conference on the Theory and Applications of Cryptographic Techniques; 1999 May 2–6; Prague, Czech Republic; 1999. p. 223–38.
[34] Commission for Energy Regulation. Smart metering trial data publication [Internet]. Commission for Energy Regulation; 2013 [cited 2019 Aug 21]. Available from: http://www.cer.ie/en/information-centre-reportsandpublications.aspx?article=5dd4bce4-ebd8-475e-b78d-da24e4ff7339.