Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 8896783, 9 pages
https://doi.org/10.1155/2020/8896783

Research Article

Lightweight Data Security Protection Method for AMI in Power Internet of Things

Wenqian Jiang,[1,2] Zhou Yang ,[1] Zhenglei Zhou,[1] and Jueyu Chen[1]

[1]*Measurement Center, Guangxi Power Grid Co., Ltd., Nanning 530010, China*
[2]*School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China*

Correspondence should be addressed to Zhou Yang; yangzhouwanshui@163.com

Received 4 September 2020; Revised 9 October 2020; Accepted 21 October 2020; Published 2 November 2020

Academic Editor: Yi-Zhang Jiang

Aiming at the security problems caused by the access of a large number of new advanced metering system (AMI) equipment and the rapid growth of new business data interaction volume and interaction frequency, a lightweight data security protection method for power Internet of things (IoT) is proposed. Firstly, based on the "cloud-edge-end" AMI system architecture, a multilevel anonymous authentication method is proposed to reduce the complexity of low-end equipment access without reauthentication when smart meters and other devices access the system. Then, when fully homomorphic encryption is used for data encryption transmission, the lightweight packet recombination protocol is introduced, the lightweight hash function is used to reduce the calculation cost, and the sliding address window mechanism is used to reduce the packet loss rate. Finally, improved secure multiparty computing (SMPC) is used to achieve frequency hopping data aggregation, using shared key to calculate local shared value for key update, reducing data interaction between massive devices and AMI cloud security server, and improving broadband utilization in data aggregation process. The experiment results indicate that the proposed method obtained better utilization in bandwidth and shorter average data collection completion time. Besides, the proposed method can ensure the information security in the interaction process.

1. Introduction

Advanced measurement infrastructure (AMI) system collects power data from smart meters on the user side, carries out data analysis and diagnosis, and realizes the state monitoring of electricity meters in the power grid and the metering of users' electricity consumption [1]. In recent years, the power Internet of things has been developed, in order to meet more business needs, such as public utilities data acquisition, distributed generation access and monitoring, charging pile data acquisition, demand side data acquisition, enterprise energy efficiency monitoring, and smart home applications. AMI system architecture is also developing towards the direction of "cloud-edge-end." The new AMI system introduces more IoT agent terminals and various types of sensor equipment. With the access of various types of equipment and the expansion of new services, the frequency and volume of data interaction between smart meters and master station systems will grow rapidly [2]. In this process, it brings a lot of challenges to the security protection. For example, the attacker can use these node devices in the system to carry out man in the middle attack, DoS attack, and data eavesdropping, steal the user's relevant privacy, tamper with the user's electricity data, and even launch a switch on attack on the user. Therefore, in the case of so many new devices and more new business data interaction, how to carry out efficient data security protection is particularly important [3, 4].

At present, homomorphic encryption and security data obfuscation are widely used to prevent attackers from monitoring users [5]. The Paillier cryptosystem is used to aggregate the data in the network to improve the ability against the internal/external attacks and protect the privacy of users [6]. However, this method has the problem of forward security of data, and the session key used is fixed. Once leaked, the security will be difficult to guarantee. The

study in [7] proposed a distributed security framework, which uses homomorphic encryption to encrypt user side data and access control in smart grid, but each data packet needs to be verified and signed. The encryption scheme has high cost and low efficiency. The study in [8] proposed a security protocol which realizes the anonymous transmission of metering data by using the way that the metering data is stored and associated with multiple watt hour meters, but this mode requires multiple encryption and decryption operations, and the utilization rate of system resources is low. In [9], a secure multiparty-based algorithm was proposed. This system can hide user data and maintain its integrity without the need of trusted third party and diagnose forged and wrong signatures. However, due to the large number of operations, the computing cost and network overhead are relatively large, so it cannot meet the application requirements of AMI.

Therefore, in order to solve the problems of low encryption efficiency and high network overhead existing in the existing methods, a lightweight data security protection method of AMI for the power Internet of things is proposed. The main innovations are as follows:

(1) The traditional main station of AMI system is under great pressure, and the data processing response efficiency is low, which cannot meet a large number of new business requirements. The proposed "cloud-edge-end" AMI system architecture introduces the edge layer, which has edge computing capability, and can give priority to the localization of end layer data in the edge layer. Under this framework, a multilevel anonymous access authentication scheme for smart meters and other devices is proposed. In the same security computing environment, the plug and play information model is used to dynamically access the edge nodes (IoT proxy terminals) to achieve smooth, low-latency, and secure service delivery from cloud to end layer.

(2) In this new architecture, full homomorphic encryption and SMPC are used to protect privacy data. Aiming at the problem of excessive fragmentation of data packets in the transmission of fully homomorphic encrypted data, lightweight packet reorganization protocol is adopted to collect data of corresponding length according to the size of packet header, which simplifies the process of packet grouping and reorganization. In order to reduce the packet loss rate, lightweight hash function is used to solve the problem of large ciphertext grouping and heavy encryption. Moreover, the sliding address window mechanism is introduced to reduce the packet loss rate.

(3) Improve the data aggregation processing of SMPC, use the shared key as the input of pseudorandom number generator in all terminal devices under the jurisdiction of each IoT agent terminal to calculate the local shared value, reduce data interaction, and improve the utilization rate of network bandwidth.

2. System Architecture and Problem Modelling

The traditional AMI architecture only has two levels: master station and smart meter. Some scenarios have concentrator in the middle layer, but it is only data forwarding. All meter data can only be uploaded to the master station for processing. The pressure of the master station is high, and the response efficiency of data processing is low, which cannot meet a large number of new business needs. The proposed "cloud-edge-end" AMI system architecture, as shown in Figure 1, introduces edge layer, has edge computing capability, and can give priority to local processing of end layer data in the edge layer.

"Cloud" refers to the cloud security server of the measurement centre, which is responsible for gathering data from edge nodes and performing advanced analysis related to big data [10]. "Edge" is the edge node, that is, the IoT agent terminal, which has a certain edge computing ability, gathers the data of devices such as smart meters, gives priority to localized processing, and sends the processing results to the cloud. "End" refers to smart meters and other types of devices (such as low-voltage fault sensor, photovoltaic inverter, charging pile acquisition device, etc.), responsible for data acquisition. In this framework, a multilevel anonymous access authentication scheme for smart meters and other devices is proposed. In the same security computing environment, the plug and play information model is used to dynamically access the IoT agent terminal to realize the dynamic management of smart meters. Combined with the network threats existing in the network data, the security goal of the system is designed, and the data security transmission in AMI system is realized through the proposed lightweight data aggregation security protection method, and the smooth, low-delay, and safe service delivery from cloud layer to end layer is realized.

2.1. System Network Model. For each side of the AMI system registration centre, there is a group of IoT proxy terminals. On the upper side, it communicates with the cloud security server of the measurement centre and generally adopts wireless public network or wireless 4G private network. On the other hand, it directly communicates with smart meters and other devices, using micro power wireless, broadband carrier, Lora, and other communication means, through single-hop plug and play connection. Each IoT agent terminal is in charge of a part of smart meters. With the help of the plug and play information model precached by IoT agent terminals, the adaptive access of smart meters can be realized. Moreover, these IoT agent terminals have edge computing capability. Through preinstalled app applications, they can independently provide services for the smart meters in charge without the help of cloud. Similarly, through the precached plug and play information model, IoT agent terminals in different places can automatically register to the cloud.

It should be pointed out that the proposed method is based on the AMI system network model and adopts multilevel anonymous access authentication method for security protection. The IoT proxy terminal and smart meter and other devices are equipped with encryption chips. When the meter is connected to the IoT proxy terminal, mutual identity authentication must be conducted. When the IoT proxy terminal is connected to the cloud security server, it must also conduct two-way identity authentication. In general, smart meters and other devices only need to mutually authenticate with IoT agent terminal and establish session key, then conduct data interaction and localization processing, and send the processing results to the cloud. This mode can reduce the frequent authentication and information interaction between the terminal devices and the master station, improve the data processing efficiency, and save the network bandwidth. Under the above AMI system architecture of the power Internet of things, the safety modelling and security protection methods are further considered. In order to facilitate the expression, the devices such as smart meters are collectively referred to as devices in the subsequent safety elaboration.

2.2. Security Modelling. In the new AMI network, the potential security risks are as follows: (1) the eavesdropper steals the data of other devices in the same network by capturing the data of a certain end device; (2) the attacker simulates the Internet of things agent terminal and sends false data collection requests to the smart meter frequently, forming a network storm and wasting network bandwidth; and (3) the eavesdropper captures and replays data packets, illegally steals electricity, and even switches on and off. Therefore, the main considerations of security modelling are as follows:

The information of the end device is protected by data encryption. Fully homomorphic encryption is used to transmit hidden data packets. Even if the eavesdropper captures the data packet, it is difficult to infer the actual reading. The actual reading can only be obtained when the private key owned by the AMI cloud security server is obtained; for the SMPC-based security protocol, the eavesdropper cannot obtaine private key. It is also necessary to know the random number generated by the target end device as the shared value of other devices before reading other device data.

Before sending the collected end device data to the AMI cloud security server, aggregate it in the network to prevent the AMI cloud security server or any third party from misusing the data. Let $q_i \forall i \in \{1, 2, \ldots, n\}$ be the reading of the end device i, encrypt it with the public key of the AMI cloud security server before transmission, that is, $\mathrm{Enc}_{\mathrm{PK}}(q_i)$, and aggregate the data of the device in the network, and the generated value y_{EG} is transmitted to the AMI cloud security server by the IoT agent terminal:

$$\sum_{i=1}^{n} \mathrm{Enc}_{\mathrm{PK}}(q_i) = y_{\mathrm{EG}}. \tag{1}$$

The same method is also applicable to SMPC-based security protocol. The hidden data can be operated by using data aggregation technology. Since the hidden data can be aggregated by SMPC-based protocol, it is unnecessary to send the actual values to the devices performing data aggregation.

Verify the identity of the sender and verify the integrity of the transmitted data. The security protocol based on SMPC uses the elliptic curve digital signature algorithm to authenticate the sender's data packet [11, 12]. The digital signature is verified by packet sender to confirm the identity of the data, and if there is no private key to create the signature, the signature cannot be forged. When the signature is invalid, the content of the data packet cannot be modified to ensure data integrity.

$$\left\{\mathrm{Enc}_{\mathrm{PK}}(q_i), \mathrm{Sig}_{\mathrm{SK}_i}(\mathrm{Enc}_{\mathrm{PK}}(q_i))\right\}. \tag{2}$$

Identify and discard duplicate messages. Since all data packets have a time stamp, if the data packet is adopted for the current data collection, then the time stamp (TS) of the data packet can be checked:

$$\left\{\langle \mathrm{Enc}_{\mathrm{PK}}(q_i), \mathrm{TS}\rangle, \mathrm{Sig}_{\mathrm{SK}_i}(\langle \mathrm{Enc}_{\mathrm{PK}}(q_i), \mathrm{TS}\rangle)\right\}. \tag{3}$$

3. Lightweight Data Aggregation Security Protection Method

The proposed method uses full homomorphic encryption to generate the public key of each IoT agent terminal. The data of the device is encrypted by the public key distributed before sending. After the data is sent to the IoT agent terminal in the region, the IoT agent terminal calculates the shared value according to the SMPC protocol and decrypts the received encrypted data if the conditions are met. After collecting the data, each IoT agent terminal encrypts and uploads the data through the key distributed by the cloud security server. The cloud security server determines whether to receive the data according to the calculated shared value and decrypts it for analysis and processing. In the above process, the lightweight packet recombination protocol is used to solve the packet recombination problem at the receiving end of the data, and the packet loss rate is reduced. When SMPC processes the key update of each terminal device, the shared key is used as the input of pseudorandom number generator to calculate the local shared value, reduce data interaction, and improve the utilization of network bandwidth [13].

3.1. Fully Homomorphic Encryption. There are two types of homomorphic encryption system: partial homomorphic encryption and fully homomorphic encryption. Partial homomorphic encryption system [14] can only perform homomorphic addition operation on ciphertext, while fully homomorphic encryption can perform homomorphic addition and multiplication for encrypted data. In this paper, fully homomorphic encryption is adopted.

In fully homomorphic encryption, Decrypt is supposed as the decryption algorithm of encryption scheme, SK is the private key, PK is the public key, $f(x_1, x_2, \ldots, x_t)$ is the $t-$ ary function, and then the decryption result of

$\text{Decrypt}(f(x_1, x_2, \ldots, x_t), \text{SK}) = f(m_1, m_2, \ldots, m_t)$ is consistent with the corresponding operation result of plaintext. Generally speaking, the fully homomorphic encryption consists of a four tuple (KeyGen, Enc, Dec, Evaluate), in which KeyGen (λ) is the key generation algorithm, which generates the required private key and public key according to the input security parameter λ; Enc(PK, m) is the encryption algorithm, which uses public key PK to encrypt plaintext m to get ciphertext c; Dec (SK, c) is a decryption algorithm, which uses private key SK to decrypt ciphertext to get plaintext; and Evaluate is a homomorphic evaluation algorithm.

The proposed method optimizes the Smart–Vercauteren (SV) scheme based on fully homomorphic encryption [15, 16]. The SV method is composed of five algorithms: KeyGen, Enc, Dec, Add, Multiply[17]. The SV method requires users to specify security parameters. The public key is composed of a prime number p and an integer $\alpha \bmod p$. Since only binary numbers are encrypted in this model, the private key is only composed of integers z.

Since the readings of the device are encrypted with public keys before being sent to the IoT agent terminal, the SV homomorphic scheme can encrypt binary numbers only. The reading R of the device every 20 minutes is converted into a fixed-point binary number Q, whose format is $Q[Q1].[QF]$, where $Q1$ and QF are integer and decimal places, respectively [18]. After the encoding is completed, the reading of each device is represented by a set of binary digits, and on this basis, it performs bitwise fully homomorphic encryption. That is, an n-bit binary number $X = (x_n, x_{n-1}, \ldots, x_1), x_i \in \{0, 1\}$, which can be encrypted with the following formula:

$$\begin{aligned} c &= (c_n, c_{n-1}, \ldots, c_1) = \text{Enc}_{\text{PK}}(X) \\ &= [\text{Enc}_{\text{PK}}(x_n), \text{Enc}_{\text{PK}}(x_{n-1}), \ldots, \text{Enc}_{\text{PK}}(x_1)]. \end{aligned} \tag{4}$$

When the IoT agent terminal receives the ciphertext c_i, it uses the decryption function Dec and the private key SK to decrypt the ciphertext c_i in the binary number x_i to obtain the binary number Q. The formula is

$$\begin{aligned} X &= (x_n, x_{n-1}, \ldots, x_1) = \text{Dec}_{\text{SK}}(c) \\ &= [\text{Dec}_{\text{SK}}(c_n), \text{Dec}_{\text{SK}}(c_{n-1}), \ldots, \text{Dec}_{\text{SK}}(c_1)]. \end{aligned} \tag{5}$$

In addition, the data encryption transmission method between the IoT proxy terminal and the cloud security server is the same as the above method, which will not be repeated.

3.2. Lightweight Packet Reorganization in Fully Homomorphic Encryption. When the SV method transmits data packets, there is a problem of excessive fragmentation of data packets. Specifically, the size of the datagram in the communications connection is controlled by the window size (WS) field in the message header. If the size of the data sent exceeds WS field, the data stream will be sent in packets, especially the large data packet ciphertext is divided into many segments to send and then recombined at the receiving end, especially in many devices after the data packet is segmented and then sent to the IoT agent. Reorganization of the terminal will consume a lot of computing overhead and increase the packet loss rate.

A lightweight packet reorganization protocol is proposed to overcome this problem, that the receiver does not know the total size of the packets it will receive. The lightweight packet reorganization protocol can make the device add a minimum header containing the size of the packet in the sender of the segment reorganization protocol. By reading the packet size and collecting the data of the corresponding size, the process of packet grouping reorganization is simplified. The lightweight hash function is used in the lightweight packet recombination protocol to reduce the computing pressure of devices in the power Internet of things, and the synchronization error between devices can be processed by sliding address window, which greatly reduces the packet loss rate of packet packets [19]. When the protocol is initialized, the initial value and key of the device IIDare transmitted to the IoT agent terminal, and the sending window value is 1, so it is not necessary to set the acceptance window; however, the IoT agent terminal will reserve the space with the window value of w for each device, and there is no need to set the sending window. When the device counter N_{now} changes and meets the address jump condition, the lightweight packet reorganization protocol is used to obtain the address of the next hop of the data packet, and the address chain list is updated; when the IoT agent terminal counter N_{now} changes, the lightweight packet reorganization protocol is used to obtain the address of the next hop of the device, and the address chain list is updated. The efficiency of communication is guaranteed by sliding address [20]. The main steps are as follows:

(1) The address is generated. In order to ensure that the rule of address hopping cannot be cracked to the greatest extent, the lightweight packet recombination protocol uses the full hash number h_i of the current address. In order to prevent the sender and receiver from being out of sync in the transmission process, the timestamp parameter t_i is not included in the hash calculation. There are

$$\begin{aligned} h_i &= H\left[h_{i-1} \| K_S\right], \\ \text{IID}_{X(i)} &= (h_i)_{0 \longrightarrow 63}, i = \{1, 2, \ldots\}, \end{aligned} \tag{6}$$

where h_i is the hash value at time t_i; the initial value h_0 of h_i is the initial IID of the device X; $\text{IID}_{X(i)}$ is the IID of the device X after the transition at time t_i; $(h_i)_{0-63}$ is the operation of fetching the first 64 bits of the data h_i; and $H[\cdot]$ is the hash calculation. When the protocol is initialized, the initial value and key of the device at the end of the interaction are transmitted to ensure data security in this link.

(2) The address link list is updated. The update of the address link list of the lightweight packet reassembly protocol is triggered by the change of the time counter N_{now}. Counter N_{now} is the number of address changes:

$$N_{\text{now}} = \lfloor \frac{T_{\text{now}} - T_0}{\Delta t} \rfloor, \tag{7}$$

where T_{now} is the time provided by the clock; T_0 is the time stamp of the initial connection of the device; Δt is the time step, that is, the time interval of address jumps; $\lfloor \cdot \rfloor$ is the data rounding down; and N_{now} is the number of address changes between the start time T_0 and the current time T_{now} of the system. When $N_{\text{now}} > N_{\text{stored}}$, make $N_{\text{stored}} = N_{\text{now}}$ and update the address list at the same time; when $N_{\text{now}} = N_{\text{stored}}$, keep the original address list state.

(3) Slide the address window. In order to solve the problems of low communication efficiency and high packet loss rate caused by unsynchronized clocks, the lightweight packet reassembly protocol uses a sliding address window mechanism. The sending address window value is 1, and the receiving address window value is w, but w will be set according to the security requirements of the session, and the address window will also slide forward with time.

3.3. Improve the Data Aggregation Processing of SMPC. In the process of data encryption and decryption in the AMI system, SMPC is used to update the keys of each device to prevent attackers from stealing or attacking other devices through one device, so as to further ensure the privacy and security of data. Among them, SMPC means that in a distributed network environment, n participants cooperate to complete a certain calculation process; that is, a key is divided into several parts and distributed to participants in a certain area, and the participants cooperate to complete the key reconstruction [21]. The process of SMPC is described as follows:

Assuming that there are n devices involved, all calculations are performed in a finite field Z_p, where p is a prime number. For the privacy secret r_i of device i, the unique point $x_i \in Z_p$ except zero is selected, and the random secret sharing polynomial $f_i(x)$ with $f_i(0) = r_i$ is selected. Its unique point x_i is sent to all other devices, and the shared value $f_j(x_i)$ calculated by the other $(n-1)$ terminal device is received, and then $F(x_i) = \sum_{K=1}^{n} f_K(x_i)$ is calculated. The above steps are completed by all devices, and the calculated $F(x_i)$ value is sent to the IoT agent terminal. The IoT agent terminal constructs a polynomial $h(x)$ of degree $(n-1)$ by using $F(x_m)$ value and Lagrange interpolation, where $m \in \{1, \ldots n\}$. The constant term of $h(x)$ is the collection of secrets of all terminal devices under the jurisdiction of the agent terminal of IoT.

Since SMPC requires communication between all devices, this will increase the complexity of communication, and the topology of the AMI network makes it infeasible [22]. For this reason, it is proposed to use a shared key in all devices under the jurisdiction of each IoT agent terminal. And the key is used as the initial input of the random number generator to preload it into the device.

Each round of data collection is initiated by the IoT agent terminal. The IoT agent terminal selects a round number c_K greater than the previous rounds, which will be sent to all devices in the network under the jurisdiction of the IoT agent terminal. Each terminal device i uses the random number generation function $\text{PRNG}_i(\cdot)$ to calculate the $f_j(x_i) = \text{PRNG}_i^{c_K}(K_j)$ value with the initial seed K_j at c_K time, where $j \in \{1, \ldots, n\}/i$. If there are n terminal devices: $\{(0, r_i), (x_1, f_1(x_i)), \ldots, (x_n, f_n(x_i))\}/(x_i, f_i(x_i))$, and a new tuple $(X_i, F_i(X_i))$ is used to represent the above points, then based on these points, the polynomial $F_i(X_i)$ of degree $(n-1)$ can be constructed. However, the coefficient of the polynomial is not a random number. And the Lagrange polynomial $l_i(x)$ must be calculated for each terminal device i to calculate the coefficient:

$$l_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}. \tag{8}$$

Then the polynomial $F_i(X)$ is

$$F_i(X) = \sum_{j=1}^{n} F_i\left(X_j\right) \cdot l_i(X). \tag{9}$$

Based on the above, the device i can use the formula in x_i instead of X to calculate its own share. After calculating all shared values, the IoT agent terminal summarizes them and uses the above method to construct a polynomial constant term on the received F_i value, which is the sum of all r_i values to complete the update of the key.

Furthermore, the AMI network under the new architecture is a multihop network, which uses the advantages of processing within the network to reduce bandwidth by using a multihop method. Specifically, the Lagrange polynomial calculated by each IoT agent terminal can be calculated by multiplying the total share calculated by all devices under its jurisdiction by the relevant Lagrange polynomial to verify the shared value under its jurisdiction. And the values are aggregated so that most of the calculation and data transmission are distributed between the IoT agent terminal and each device. Finally, the IoT agent terminal only needs to sign the result and send it to the AMI cloud security server, which will further reduce the data interaction between the mass terminal equipment and the AMI cloud security server, improve the broadband utilization rate in the data aggregation process, and reduce calculations overhead.

4. Simulation Results and Analysis

Network simulator ns-3 [23] is applied to evaluate the performance of the proposed method, and a "cloud-edge-end" multihop network topology is established, which is composed of 50 random multihop edge end local networks with the size of N. For each local network, a grid node is used as the agent terminal of IoT, and the $(N-1)$ grid node is used as the device. Set the device data to 32 bits, global clock synchronization, with time stamp.

The method in [7] and [9] is compared with the proposed method in simulation system, and the performance of

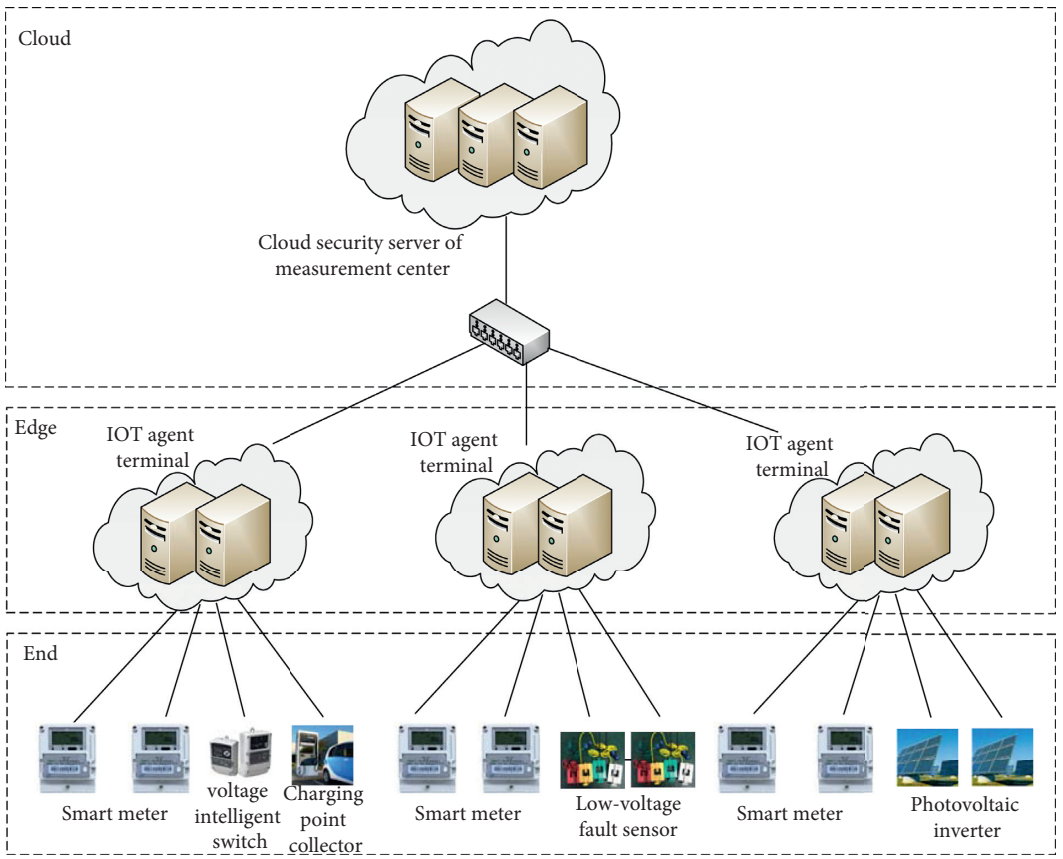

Figure 1: AMI system architecture based on "cloud-edge-end."

each encryption method is analyzed. Set the random number generator for data aggregation to generate 128-bit random numbers for identity verification. The data packet transfer rate, throughput, and the average data collection completion time are adopted as the comparison index. The data packet transfer rate is the ratio of the number of data packets received by the IoT agent terminal to the number of data packets sent by the device. The throughput is the total amount of data received by the IoT agent terminal per second. And the average data collection completion time is the average time it takes for the IoT agent terminal to collect a round of device data.

4.1. Packet Transfer Rate. Figure 2 shows the data packet transfer rate of each method for different devices.

It can be seen from Figure 2 that for all methods, the packet transfer rate is almost 100%, until after the 145-device topology, the packet transfer rate of the methods of [7] and [9] begins to decline. Compared with the proposed method, other methods generate larger data packets, and the larger the data volume, the higher the probability of congestion. In general, the increase in the number of devices will not significantly reduce the packet transfer rate performance of the proposed method.

4.2. Throughput. The throughput of each method is shown in Figure 3.

It can be seen from Figure 3 that as the number of devices in the network increases, the throughput values of different methods also increase. The throughput value is generated according to the size of the generated data packet, and the proposed method produces the smallest throughput. It can be seen that compared with other methods, it generates the smallest data packet. Through throughput analysis, it can be seen that the proposed method uses the least network bandwidth.

4.3. Average Data Collection Completion Time. The average data collection completion time is an important indicator of some AMI applications. The average data collection completion time value of each method is shown in Figure 4.

It can be seen from Figure 4, with the growth of the network, the average data collection completion time value of each method will increase. Compared with [7] (60s/120s), it takes less time for [9] and the proposed method to complete a round of data collection. That is why the size of data packets generated by Paillier cryptosystem and random number generator is much smaller than that generated by SV method. With the increase of data, the data is divided into smaller packets according to the size of the window SV method, which increases the possibility of collision when accessing the channel to transmit data. Each conflict will increase the waiting time of avoiding collision and increase the reorganization cost, thus increasing the collection completion time. In addition, the average data collection completion time of the proposed method is slightly larger than that of the Paillier cryptosystem,

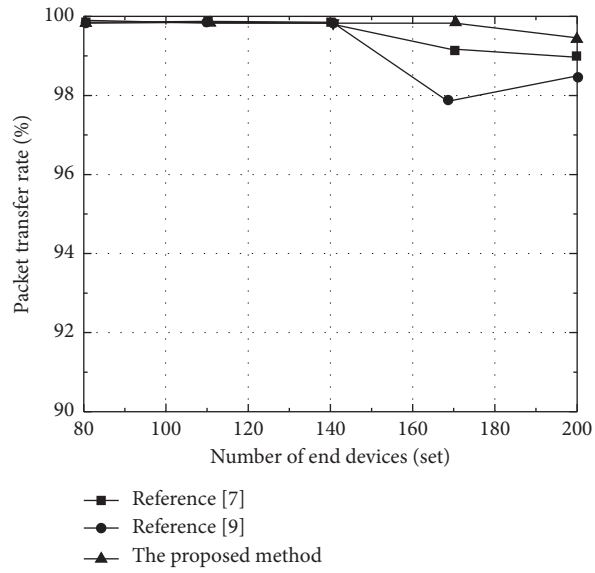

Figure 2: Packet delivery rate of each method with different number of devices.

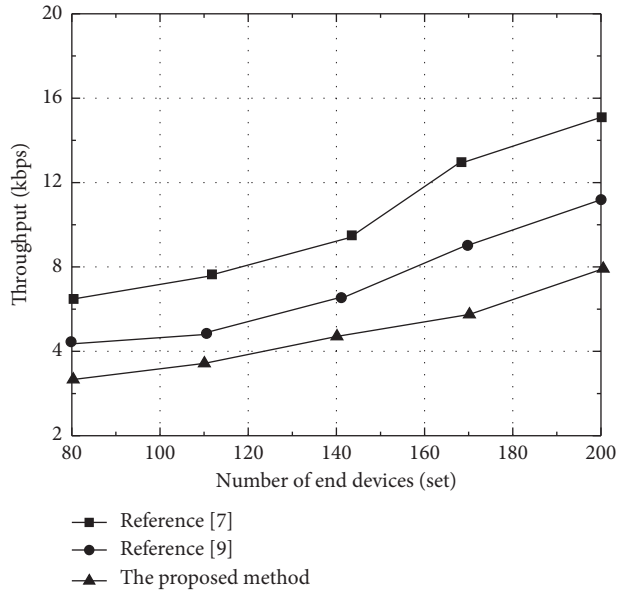

Figure 3: Throughput value of each method with different number of devices.

which is due to the addition of an improved SMPC key update mechanism to ensure the security defense between devices. Although it increases a small part of the computational overhead, the overall performance of the proposed method is still better than the other two methods.

4.4. Security Analysis. In order to demonstrate the data security of the proposed method, it is compared with the methods in [7] and [9], and the results are shown in Figure 5. The security index uses the ratio of the number of malicious attacks and the total number of malicious attacks.

It can be seen from Figure 5 that with the increase of network attacks, the security index of [7] begins to decline slowly. After the number of attacks increases in the later period, the index decreases faster, and the ability to resist network threats is weak. The security index of [9] drops rapidly, and the security performance drops sharply when there are a large number of network attacks. These two methods do not have the ability to resist many network attacks. With the increase of network attack, the proposed model decreases smoothly and tends to be stable, so it is suitable for the current network data security requirements under the environment of power Internet of things.

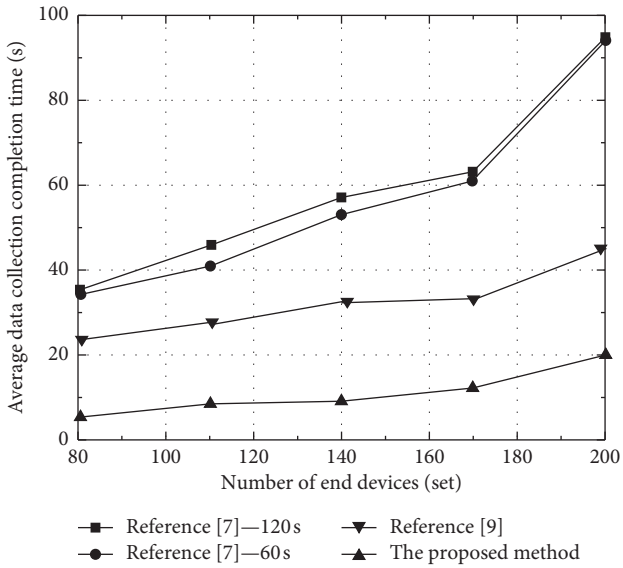

Figure 4: The average data collection completion time value of each method with different number of devices.

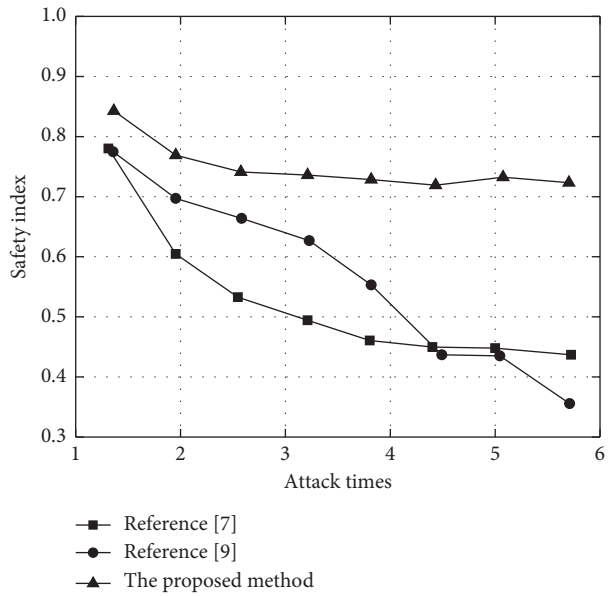

Figure 5: Security comparison of different methods.

5. Conclusions

In order to ensure the reliability and privacy of the new AMI system data in the power Internet of things, the proposed method is based on the "cloud-edge-end" AMI system architecture and proposes a lightweight data aggregation protection method using fully homomorphic encryption and SMPC. In order to solve the problem that the aggregate data size of SV method changes, a packet reorganization protocol is proposed to reduce the cost of data packet reorganization process. In SMPC, the secret sharing technology of pseudorandom number generator is used to hide data information through data aggregation, which improves the security of data information. In addition, the performance of the proposed method is demonstrated. The experimental results show that the proposed method performs better than the traditional SV method in terms of bandwidth utilization and data collection. The proposed method is feasible for AMI system security protection based on "cloud-edge-end" architecture.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Guangxi Power Grid Co., Ltd., Science and Technology Project (GXKJXM20170393) and Key Technology Research and Application of New Generation Advanced Measurement Infrastructure.

References

[1] Y. C. Li, P. Zhang, and S. Q. Zheng, "Power data privacy protection based on empirical mode decomposition and homomorphic encryption," *Power System Technology*, vol. 43, no. 5, pp. 1810–1818, 2019.

[2] C. Xu, J. Ren, D. Zhang, and Y. Zhang, "Distilling at the edge: a local differential privacy obfuscation framework for IoT data analytics," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 20–25, 2018.

[3] A. Sundararajan, T. Khan, A. Moghadasi, and A. I. Sarwat, "Survey on synchrophasor data quality and cybersecurity challenges, and evaluation of their interdependencies," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 3, pp. 449–467, 2019.

[4] Y. C. Li, R. X. Qiu, and J. Zeng, "Smart grid blind online false data injection attack based on nuclear principal component analysis," *Power System Technology*, vol. 42, no. 7, pp. 2270–2278, 2018.

[5] A. Tajer, S. Sihag, and K. Alnajjar, "Non-linear state recovery in power system under bad data and cyber attacks," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 5, pp. 1071–1080, 2019.

[6] C. Lin, D. He, N. Kumar, K.-K. R. Choo, A. Vinel, and X. Huang, "Security and privacy for the internet of drones: challenges and solutions," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 64–69, 2018.

[7] S. Kulkarni, Q. Gu, E. Myers et al., "Enabling a decentralized smart grid using autonomous edge control devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7406–7419, 2019.

[8] T. Zhang, Dy Zhao, F. Xue et al., "Research framework of information security protection technology for intelligent terminals in power system," *Automation of Electric Power Systems*, vol. 43, no. 19, pp. 1–8, 2019.

[9] M. J. Collins, "Efficient secure multiparty computation of sparse vector dot products," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 21, no. 5, pp. 1107–1117, 2018.

[10] J. Lu, W. P. Luan, R. L. Liu et al., "Distribution Internet of things architecture based on comprehensive perception and software definition," *Power System Technology*, vol. 42, no. 10, pp. 3108–3115, 2018.

[11] H. Qiu, M. M. Qiu, and Z. Ming, "Lightweight selective encryption for social data protection based on EBCOT coding," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 1, pp. 205–214, 2020.

[12] X. Liu, "Cloud computing security scheduling scheme based on jitter strategy optimization," *Scientific Journal of Control Engineering*, vol. 25, no. 5, pp. 889–896, 2018.

[13] A. Jolfaei and K. Kant, "A lightweight integrity protection scheme for low latency smart grid applications," *Computers & Security*, vol. 86, no. 9, pp. 471–483, 2019.

[14] K. Xu, W. Zhang, and Z. Yan, "A privacy-preserving mobile application recommender system based on trust evaluation," *Journal of Computational Science*, vol. 26, no. 5, pp. 87–107, 2018.

[15] S. Tonyali, K. Akkaya, N. Saputro, A. S. Uluagac, and M. Nojoumian, "Privacy-preserving protocols for secure and reliable data aggregation in IoT-enabled smart metering systems," *Future Generation Computer Systems*, vol. 78, no. 3, pp. 547–557, 2018.

[16] J. Shen, H. D. Zou, and W. B. DaiYuan, "A domain-divided configurable security model for cloud computing-based telecommunication services," *The Journal of Supercomputing*, vol. 75, no. 1, pp. 109–122, 2019.

[17] M. S. Rahman, I. Khalil, A. Alabdulatif, and X. Yi, "Privacy preserving service selection using fully homomorphic encryption scheme on untrusted cloud service platform," *Knowledge-Based Systems*, vol. 180, no. 13, pp. 104–115, 2019.

[18] K. Fan, H. W. Jiang, and Y. Yang, "Lightweight RFID protocol for medical privacy protection in IoT," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1656–1665, 2018.

[19] J. J. Zhen, Z. L. Ying, Y. H. Zhao et al., "Application of deep learning and iterative quantization in image retrieval," *Journal of Signal Processing*, vol. 5, no. 35, pp. 919–925, 2019.

[20] Q. Feng, H. D. He, and K.-K. R. L. ChooZhou, "Lightweight collaborative authentication with key protection for smart electronic health record system," *IEEE Sensors Journal*, vol. 20, no. 4, pp. 2181–2196, 2020.

[21] X. Yan, Y. Lu, L. Liu, and D. Ma, "Image secret sharing construction for general access structure with meaningful share," *International Journal of Digital Crime and Forensics*, vol. 10, no. 3, pp. 66–77, 2018.

[22] A. Sahi, D. Lai, and Y. Li, "Three-party password-based authenticated key exchange protocol based on the computational Diffie-Hellman assumption," *International Journal of Communication Networks and Distributed Systems*, vol. 21, no. 4, pp. 560–581, 2018.

[23] Z. Hossain, Q. Xia, and J. M. Jornet, "TeraSim: an ns-3 extension to simulate Terahertz-band communication networks," *Nano Communication Networks*, vol. 17, no. 3, pp. 36–44, 2018.