Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8810393, 10 pages
https://doi.org/10.1155/2020/8810393

*Research Article*

# Fault-Tolerant Privacy-Preserving Data Aggregation for Smart Grid

**Huadong Liu**,[1,2] **Tianlong Gu**,[2] **Yining Liu**,[2] **Jingcheng Song**,[2] **and Zhixin Zeng**[2]

[1]*School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi 710071, China*
[2]*Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China*

Correspondence should be addressed to Huadong Liu; ldd@guet.edu.cn and Yining Liu; lyn7311@sina.com

Received 23 July 2020; Revised 3 September 2020; Accepted 14 September 2020; Published 30 September 2020

Academic Editor: Weizhi Meng



In smart grids (SG), data aggregation is widely used to strike a balance between data usability and privacy protection. The fault tolerance is an important requirement to improve the robustness of data aggregation protocols, which enables normal execution of the protocols even with failures on some entities. However, to achieve fault tolerance, most schemes either sacrifice the aggregation accuracy due to the use of differential privacy or substitution strategy or need to rely on an online trusted entity to manage all user blinding factors. In this paper, a $(k, n)$ threshold privacy-preserving data aggregation scheme named $(k, n)$-PDA is proposed, which reconciles data usability and data privacy through the BGN cryptosystem and achieves fault tolerance with accurate aggregation using Shamir's secret sharing without any online trusted entity. Besides, our scheme supports the efficient changing of users' membership. Specifically, the dynamic secrete key is distributed to $n$ smart meters (SMs) through the threshold secret sharing algorithm. When $k$ or more meters participate in the aggregation, the data service center (DSC) can reconstruct the key to compute the aggregate results, and less than $k$ SMs cannot recover the key. Thus, our solution still works functionally even if up to $n - k$ SMs fail; also, it resists attacks from the collusion of less than $k$ SMs. Moreover, system and performance analyses demonstrate that our scheme achieves privacy, fault tolerance, and membership dynamics with high efficiency.

## 1. Introduction

The development of information, communication technology, and advanced control technology has driven the emergence of the smart grid. In SG, the sophisticated control system uses the real-time electricity consumption data monitored from smart meters to balance the supply and demand of electricity, thereby to stabilize power supply and improve power quality. However, fine-grained consumption data may pose a threat to consumers' privacy. Some researchers have pointed out that according to the real-time electricity consumption data, data collectors or eavesdroppers can infer consumers' living habits, household occupancy, economic conditions, or even which appliances are being used [1–3]. If the real-time consumption data is collected without assurance of users' privacy, the smart grid would be hardly developed. Therefore, how to ensure the usability of data while protecting the privacy of users has become a concern of researchers [3, 4]. For the control system of a smart grid, it is sufficient to know the total instantaneous power demand and the power supply in a certain area. So, among the popular solutions is data aggregation which provides the sum of the real-time consumption data of users in a group rather than the data of each user [3–5].

Researchers have proposed many efficient privacy-preserving data aggregation protocols which can be classified into two types: fault-intolerant schemes and fault-tolerant schemes. In fault-intolerant schemes [6–10], the system can carry out the scheme to obtain aggregate data with privacy-preservation when all entities work well. However, the aggregation process may be stopped due to failures on smart meters. As a continuously operating system, the smart grid cannot be completely fault-free. Therefore, this type of aggregation scheme is impractical. In fault-tolerant schemes [11–17], the system can still work and compute the aggregation result despite some failures on SMs. Specifically, some fault-

tolerant schemes are based on differential privacy, replacement strategies, but the aggregation result is an approximate value; others are based on an online trusted entity to manage blind factors of all SMs and the aggregator, which may increase the privacy risk. However, accurate aggregate values are the basis for the smart grid to accurately grasp real-time loads.

In addition to fault tolerance, dynamic membership management is also very important to the practicality of smart grids. In schemes that have no consideration on the dynamic management of members, any changing of membership, withdrawal, or joining may even cause all the entities in the system to be reconfigured. At the same time, a large amount of computation and communication will be imposed on the system. However, both the migration of users and the alternation of power providers will lead to changes in membership.

In this paper, we propose a novel privacy-preserving data aggregation protocol named $(k, n)$-PDA in smart grids where $n$ is the number of *SMs* in the aggregation area, and $k$ is the threshold. Our solution is based on the BGN cryptosystem and Shamir's secret sharing algorithm. The main contributions of this paper are summarized as follows:

(i) We construct the encryption, aggregation, and decryption process based on the BGN homomorphic cryptosystem to ensure the confidentiality and privacy of data

(ii) We use the threshold characteristics of Shamir's secret sharing algorithm to make the aggregation scheme threshold fault-tolerant, which means that accurate aggregate value with privacy preservation can be obtained even when $n$-$k$ SMs collude with each other or do not work normally. The threshold $k$ can be set according to experience and security to avoid the system still performing meaningless aggregation when a serious abnormality happens in the grid. Moreover, Shamir's algorithm makes our scheme easily achieve dynamic membership management

(iii) We use the one-time pad to achieve forward security

(iv) We analyze the security and some other system properties to show that the proposed scheme holds confidentiality, privacy preservation, fault tolerance, dynamic membership, forward security, and no need for any online trusted or high authority entity. Also, we evaluate the efficiency of the system to confirm that our solution has a good real-time performance

The rest of this paper is organized as follows: Section 2 introduces some related works. We present some preliminaries in Section 3. The system model, adversarial model, and design goals are described in Section 4, and our scheme is detailed in Section 5. System analysis and performance evaluation of the scheme are shown, respectively, in Section 6 and Section 7. Section 8 concludes this article.

## 2. Related Works

The communication security and data privacy protection in the smart grid have received great attention from researchers. Many excellent solutions have been proposed to ensure communication security through methods such as authentication or key management [18–20]. In terms of privacy protection, the popular methods are anonymization and data aggregation. The anonymization scheme [21] delinks individual raw data and their source. However, attackers may relink the raw data and the source by depseudonymization [22, 23].

In recent years, many effective data aggregation schemes have been proposed to aggregate data of consumers with privacy preservation. Some of them cannot run when any part of entities fails to work, and others are fault-tolerant.

Schemes that are intolerant of fault, such as [7–10], are usually based on a group of random integers which sum to zero. These random integers are distributed to SMs and the aggregator as blind factors. SMs use blind factors to mask their data to achieve privacy and encrypt the masked data with homomorphic encryption. Then, the aggregator removes these masking factors from the aggregate ciphertext with its private key to obtain aggregate ciphertext. Finally, the data service center (DSC) decrypts the ciphertext to get accurate aggregate values. However, if any SM cannot send information to the aggregator (AG), AG cannot eliminate the blinding factor from the aggregate ciphertext. Consequently, DSC cannot obtain the aggregate value.

To achieve fault tolerance, schemes with fault tolerance usually adopt differential privacy, substitution strategy, centralized management of user blinding factors, etc., or a combination of two or more of them.

Schemes based on differential privacy [11, 12] mask the original data by adding random noises that follow a randomized function distribution. Finally, these noises are removed according to the expected value of the function from the aggregate data to get an approximation of the aggregation result.

Substitution strategy [13–15] is that the faulty users' data are replaced with the data of other users who have the same blinding factors. In [15], if an SM, such as $SM_i$, in a group fails to send the message, the data aggregation device will select an SM, such as $SM_j$, from other groups with the same blinding factor, to replace the malfunctioning $SM_i$ so that the aggregation process can proceed. To reduce the error, this kind of schemes usually processes the data of $SM_j$ based on the past data of the group. Although these two kinds of schemes are fault-tolerant, they cannot provide accurate aggregate results.

The schemes with centralized management of blinding factors [14, 16, 17] essentially require an online entity or a trusted authority to manage blinding factors for the aggregator and the smart meters. In FESDA [16], the control center (CC) keeps all users' blinding factors. When some users fail to participate in the aggregation, CC calculates the sum of all the blinding factors of the failed users. Then, the sum is used to decrypt the aggregate ciphertext to obtain accurate results. In PDAFT [14] and PPFA [17], when an SM fails to upload data, the trusted third party will send the blind factor of the SM to help the aggregation process. However, the centralized management of blinding factors needs an online trusted authority or an online entity with high authority to hold all blinding factors, which will bring risks to users' privacy.

Recently, some fault-tolerant aggregation schemes without any trusted authority have been proposed [24, 25]. In [24], K. Xue et al. use the $(t, n)$ threshold secret sharing scheme to achieve flexible dynamic user management. In detail, each SM in building area networks (BAN) has a secret key, and the sum of these keys is zero. Every user needs to choose randomly a group of users to share its key with the secret sharing algorithm. When an SM fails, the control center (CC) needs to broadcast the identity of the failed SM and collect enough shares to recover its secret key. If the fault SM is restored, it needs to generate a new key and shares the key to a newly chosen group of users through a secure channel, which is impractical for a continuously running system. In [25], Wang et al. proposed to use multiple subsets and blinding factors to achieve privacy-preserving data aggregation. Users negotiate to update the blinding factor. If some SMs are fault, the aggregator (AG) publishes the event and their identities. Then, their cooperators have to remove their parts from the blind factors and execute the encryption again. Finally, all normal users need to report their ciphertext again. These solutions can obtain accurate aggregate values. However, they require a complex mechanism to deal with SMs' malfunction, which may cause a heavy computation and communication burden to the system.

## 3. Preliminaries

In this section, we briefly review some important algorithms that are used as the building of our scheme.

*3.1. Bilinear Map of Composite Order Groups.* A bilinear map of composite order groups related to an inputting security parameter $\tau \in \mathbb{Z}^+$ is defined as a 5-tuple $(p, q, G, G_1, e)$, where $p, q$ are two random $\tau$-bit primes, $G, G_1$ are two multiplicative cyclic groups of order $N = pq$, and $e$ is a bilinear map $e : G \times G \longrightarrow G_1$ with the following properties:

(1) *Bilinearity*: $\forall u, v \in G$, and $\forall a, b \in Z_N$, we have $e(u^a, v^b) = e(u, v)^{ab}$

(2) *Nondegeneracy*: $g$ is a generator of G; then, $e(g, g)$ must be a generator of $G_1$ and $e(g, g) \neq 1_{G_1}$

(3) *Computability*: $\forall u, v \in G$, there is a polynomial-time algorithm to calculate $e(u, v)$.

*3.2. Boneh-Goh-Nissim Cryptosystem.* The Boneh-Goh-Nissim (BGN) cryptosystem [26] is a homomorphic encryption scheme supporting unlimited addition operations but at most one multiplication and is widely applied in privacy-preserving computation. It consists of three phases: key generation, encryption, and decryption.

(1) *Key generation*: given a security parameter $\tau \in Z^+$, the algorithm $\mathcal{G}$ outputs a bilinear map of composite order groups $(p, q, G, G_1, e)$ as described in Billinear Map of Composite Order Groups. The key management agency chooses two random generators $g, u$ of $G$ and calculates $h = u^p$ and $N = pq$, where $h$ is a generator of the subgroup of $G$ with order $q$. As a result, we have public key $PK = \{N, G, G_1, e, g, h\}$ and private key $SK = q$

(2) *Encryption*: for a message $m \in Z_T$, $T < p$ picks a random $r \in Z_{N-1}$ and encrypts $m$ into $C = g^m h^r \in G$

(3) *Decryption*: with private key $SK = q$ to decrypt the ciphertext $C$, calculate $C^q = (g^m h^r)^q = g^{mq} = (g^q)^m$. Let $\widehat{g} = g^q$, then $m$ can be recovered by using Pollard's lambda method [27] to solve the discrete logarithm of $C^q$ base $\widehat{g}$ with time complexity $O(\sqrt{T})$

*3.3. Shamir's Secret Sharing Scheme.* Shamir's secret sharing scheme [28] is a $(k, n)$ threshold secret sharing scheme where a secret $S$ is divided into $n$ pieces, and the secret $S$ can be easily calculated when $k$ or more secret shares are known, while it is impossible to reconstruct $S$ if the number of known secret shares are less than $k$. In the scheme, all elements are in a limited field. We can realize this scheme in a limited field of size $P$, where $P$ is a prime number, by constructing a polynomial:

$$y = f(x) = S + q_1 x + q_2 x^2 + \cdots + q_{k-1} x^{k-1} \mod P, \quad (1)$$

where $q_1, q_2, \cdots, q_{k-1}$ are random integers less than $P$, and each participant gets a unique $x_i$ and calculates $y_i = f(x_i)$; then, $(x_i, y_i)$ is a secret share. With $k$ different shares, the polynomial can be reconstructed by the Lagrange interpolation as below:

$$f(x) = \sum_{j=1}^{k} \left( y_j \prod_{i=1, i \neq j}^{k} \frac{x - x_i}{x_j - x_i} \right). \quad (2)$$

However, we just need to find $S$ from the polynomial $f(x)$ . $S$ is the free coefficient. So, we only need to calculate

$$S = f(0) = \sum_{j=1}^{k} \left( y_j \prod_{i=1, i \neq j}^{k} \frac{x_i}{x_i - x_j} \right). \quad (3)$$

## 4. System Setup

The $(k, n)$-PDA scheme includes four entities in the system and targets to get several design goals; meanwhile, against some attacks which may be launched by entities defined as the adversarial model in Adversarial Model and Assumptions. In this section, we introduce the system model formally and describe the adversarial model and design goals in detail.

*4.1. System Model.* Figure 1 shows the system model, which consists of four kinds of entities.

(1) *Smart meters* (SMs): The SMs are devices equipped in energy consumers' houses to collect users' real-time energy consumption in every sampling time (like 15 minutes), encrypt these data, and send them to the aggregator at the end of every time slot. Usually, customers are grouped according to their locations. In this paper, we assume each aggregation domain

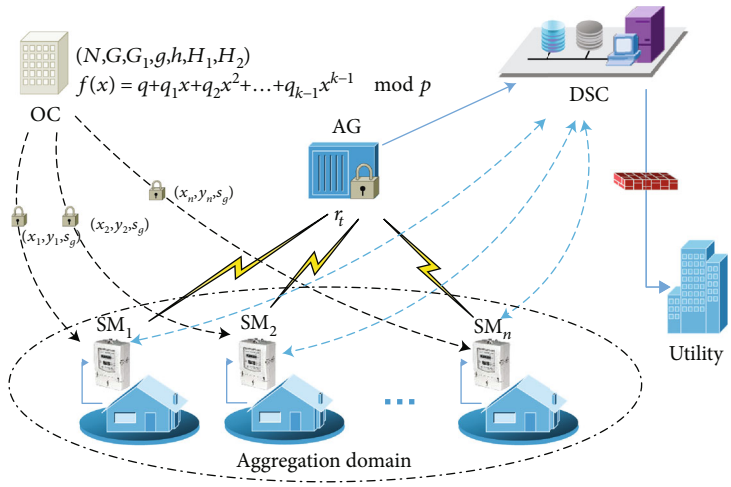


Figure 1: System model.

has $n$ SMs recorded as a set $U_g = \{SM_1, SM_2, \cdots, SM_n\}$. It should be noted that some SMs may be faulty and cannot take part in the aggregation. Suppose there are $l$ SMs that are online and participating in the aggregation, these SMs make up $U_{on} \subseteq U_g$ and $|U_{on}| = l$. We record the set of offline SMs as $U_{off} = U_g - U_{on}$ and $|U_{off}| = n - l$

(2) *Aggregator* (AG): AG is used to verify the identities of SMs, sum encrypted data from online *SMs* in the same group, and send the sum to the data service center

(3) *Data service center* (DSC): DSC decrypts ciphertexts received from legal aggregators and gets the sum of the consumption data of the set $U_{on}$ at each time slot

(4) *Operating Center* (OC): the operating center provides user registration services and key initialization for the smart grid

*4.2. Adversarial Model and Assumptions.* In our system, OC is a trusted organization. DSC and AG are faithful to performing system tasks but are curious to know the users' real-time data. Every SM (customer) will try its best to protect the privacy of data and may also infer private data of other customers through public information and its private data. An external attacker may monitor the communication channels and try to figure out users' sensitive data.

Since there are many excellent solutions [18–20] to ensure communication security in the smart grid and this paper mainly focuses on the privacy protection of users, we assume that all transmitted messages in the system are properly authenticated with existing signature methods to achieve the required authentication and integrity. We also assume all physical participants are tamper-proof and sealed, so that any illegal reading from physical devices will be perceived, and any alteration to data from entities cannot be achieved without being detected.

*4.3. Design Goals.* Our solution aims to provide aggregate data without revealing users' private data. At the same time, on the premise of ensuring confidentiality, to make this protocol more practical, we hope that when some smart meters fail to send their messages, the system can still aggregate the remaining users' data. Besides, when a user joins in or logs out, we hope that other users are not affected. Our design goals are detailed as follows:

*Confidentiality:* external attackers may eavesdrop on the messages transmitted on the communication channel. It should be ensured that unauthorized entities cannot obtain any useful information from these messages.

*Privacy:* the aggregate data is available to the public utilities; meanwhile, the individual data of every customer cannot be obtained by any other entities.

*Fault tolerance:* if any fault on SMs may cause data aggregation to fail, the usefulness of the system will be greatly reduced. Therefore, we are committed to designing an aggregation protocol that works well when even $n - k$ SMs cannot normally send consumption data, where $n$ is the total number of *SMs* in a group and $k$ is the threshold number of SMs working normally.

*Dynamic membership:* when a new user joins or an old user logs out, the system should not need to update any parameter of other users.

*Forward security:* in order to improve the antirisk ability, it is required that even if the current key is compromised, the adversary cannot find out the previous individual data.

## 5. Our Scheme

This section presents our privacy-preserving data aggregation protocol. The procedure includes four phases: system initialization, encryption, data aggregation, and decryption. In the initialization phase, OC generates and publishes system parameters and registers for SMs. In the encryption phase, AG publishes a random number to SMs; then, each SM generates a dynamic key to encrypt real-time readings and report the ciphertexts to AG; also, each SM computes

its share of the dynamic key and sends to DSC. In the aggregation phase, AG aggregates the received ciphertexts and reports the aggregate ciphertext to DSC. Finally, in the decryption phase, DSC reconstructs the dynamic key and decrypts the aggregate ciphertext to obtain the plaintext of aggregate data. The frame of the proposed scheme is shown in Figure 2 and notations to be used in the rest of the paper are listed in Table 1.

*5.1. System Initialization.*

$$y = f(x) = q + q_1 x + q_2 x^2 + \cdots + q_{k-1} x^{k-1} \bmod P. \quad (4)$$

*Step 1.* with the algorithm $\mathcal{G}$ and the input secure parameter $\tau \in Z^+$, OC generates a bilinear map of composite order groups $(p, q, G, G_1, e)$ and computes $N = pq$.

*Step 2.* OC chooses two generator $g, u$ of $G$ and gets $h = u^p$.

*Step 3.* OC selects a prime number $P$ greater than $n$ and chooses a secure argument $k$ as the threshold based on data privacy and failure rate. Specifically, the higher the failure rate is, the smaller $k$ should be, but too small $k$ will affect the user's privacy or arouse a meaningless aggregation when a serious abnormality happens, so a good balance needs to be struck. Notably, these parameters satisfy $1 < k < n < P$.

*Step 4.* OC gets $k-1$ random numbers $q_1, q_2, \cdots, q_{k-1}$ with $q_i < P$ and constructs a Shamir secret sharing model with $q$ as the secret:

*Step 5.* if a user has registered successfully, OC chooses a unique $x_i$ as the user's identity (ID) and evaluates $y_i = f(x_i)$. Also, OC generates a random number $s_g \in Z_N^*$ as the group key for users in the same group and then sends $(x_i, y_i, s_g)$ to the user through a safe channel (usually embedded in the SM and cannot be read by external devices).

*Step 6.* OC chooses two secure hash functions $H1, H2 : \{0,1\}^* \longrightarrow Z_N^*$.

*Step 7.* We use $m_i$ to represent the reading of $SM_i$. OC chooses a positive integer $T$ according to the upper limit of consumption data, satisfying $0 \le m_i < T < P < p$.

*Step 8.* OC publishes $(N, G, G_1, e, g, h, H1, H2)$.

*Step 9.* DSC produces a secret/public key pair $(\mathrm{sk}_{\mathrm{DSC}}, \mathrm{pk}_{\mathrm{DSC}})$ based on the RSA cryptosystem and releases $\mathrm{pk}_{\mathrm{DSC}}$ to SMs. DSC also selects a unique ID for the aggregator, marked as $\mathrm{ID}_{\mathrm{AG}}$.

*5.2. Encryption.* Usually, AG collects users' data every 15 minutes and aggregates them. Users encrypt their private data before forwarding them to AG. The following are the detailed steps of the encryption process at time $t$.

$$C_i = g^{\frac{m_i}{H}} h^{\frac{r_i}{H}} \quad (5)$$

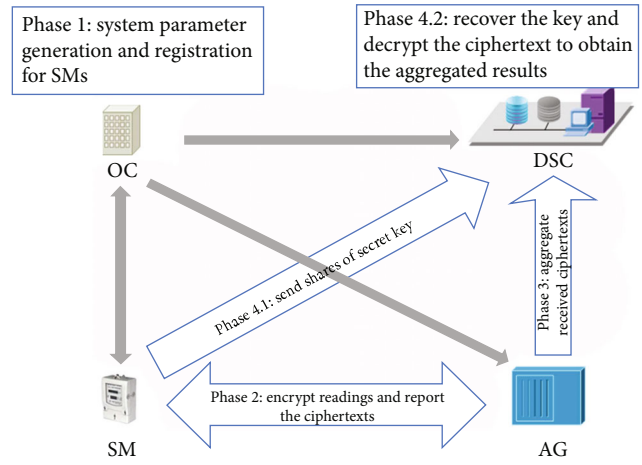


Figure 2: Frame of the proposed scheme.

Table 1: Notations.

| Symbol | Definition |
|---|---|
| SM | Smart meter |
| AG | Aggregator |
| DSC | Data service center |
| OC | Operating center |
| $U_g$ | The set of all SMs in a group |
| $U_{\mathrm{on}}$ | The set of online SMs in $U_g$, $U_{\mathrm{on}} \subseteq U_g$ |
| $n$ | $n = \lvert U_g \rvert$ |
| $k$ | The threshold of the secret sharing scheme |
| $G, G_1$ | Multiplicative group |
| $p, q$ | Two big prime number with the same length |
| $N$ | $N = pq$, the order of $G$ |
| $H1, H2$ | Secure one-way hash function: $\{0,1\}^* \longrightarrow Z_N^*$ |
| $g, u$ | Generators of $G$ |
| $(\mathrm{sk}_{\mathrm{DSC}}, \mathrm{pk}_{\mathrm{DSC}})$ | The secret/public key pair of DSC |
| $x_i$ | The identity of user $i$ |
| $s_g$ | The group key chosen by OC |
| $r_t$ | A random number generated by AG at time $t$ |
| $m_i$ | The real-time reading of $\mathrm{SM}_i$ at time $t$ |
| $C_i$ | The ciphertext of $m_i$ |

*Step 1. AG* generates and publishes a random number $r_t \in Z_p^*$ to SMs of the same group before data collection.

*Step 2.* $\mathrm{SM}_i$ $(\mathrm{SM}_i \in U_{\mathrm{on}})$ computes the dynamic secret key $H = H1(r_t | t | s_g)$, generates a random number $r_i$, and then encrypts $m_i$ with $H$ and $r_i$ to generate the ciphertext $C_i$ as equation (5).

*Step 3.* $SM_i$ generates the signature $\sigma_{i\mathrm{AG}}$ for $H2(x_i | t | C_i)$ and sends $(x_i, \sigma_{i\mathrm{AG}}, C_i)$ to AG.

*Step 4.* $SM_i$ encrypts $H \cdot y_i$ with the public key $\mathrm{pk}_{\mathrm{DSC}}$ to generate ciphertext $C_{i\mathrm{DSC}} = E_{\mathrm{DSC}}(H \bullet y_i)$ and a signature $\sigma_{i\mathrm{DSC}}$ for $H2\,(C_{i\mathrm{DSC}}|t|x_i)$, then sends $(x_i,\ \sigma_{i\mathrm{DSC}},\ C_{i\mathrm{DSC}})$ to *DSC*. To ensure better real-time, this information can be sent out at the idle time before the decryption stage.

*5.3. Data Aggregation.* After receiving messages from $SM_i$, *AG* verifies the signature $\sigma_{iAG}$ first. If the verification fails, the message will be discarded. If *AG* confirms that there are $l(\mathrm{n} \geq l \geq k)$ legitimate users sent their data, the aggregation processes as follows.

$$C = \prod_{i=1}^{l} C_i = g^{\sum_{i=1}^{l} \frac{m_i}{H}} h^{\sum_{i=1}^{l} \frac{r_i}{H}} \tag{6}$$

*Step 1.* AG aggregates all ciphertexts from all online SM*s* to obtain the aggregate ciphertext *C*.

*Step 2.* AG generates a signature $\sigma_{\mathrm{AG}}$ for $H2(C|t|\mathrm{ID}_{\mathrm{AG}})$ and sends $(\sigma_{\mathrm{AG}}, C, \mathrm{ID}_{\mathrm{AG}})$ to DSC.

*5.4. Decryption.* Firstly, DSC verifies the identity of SMs and AG and confirms the number *l* of SM*s*. Secondly, the DSC needs to obtain the secret key from the messages of SM*s* to decrypt the ciphertext *C* sent by AG. Therefore, the decryption process is divided into two steps: reconstructing the secret key and decrypting *C*.

*5.4.1. Key Reconstruction. DSC* decrypts the $C_{i\mathrm{DSC}}$ from $\mathrm{SM}_i$ $(\mathrm{SM}_i \in U_{\mathrm{on}})$ with the private key $\mathrm{sk}_{\mathrm{DSC}}$ to obtain $H \cdot y_i$ and then uses $H \cdot y_i$ of *k* SM*s* to construct equation (7) by the Lagrange interpolation.

$$H \bullet f(x) = H \bullet q + H \bullet q_1 \bullet x + H \bullet q_2 \bullet x^2 + \cdots + H \bullet q_{k-1} \bullet x^{k-1} \mod P. \tag{7}$$

Compute $H \bullet q = \sum_{j=1}^{k} (H \bullet y_j \prod_{i=1, i \neq j}^{k} (x_i / x_i - x_j))$ according to equation (3).

*5.4.2. Decrypting C. DSC* decrypts *C* with $H \cdot q$.

$$C^{H \cdot q} = \left( g^{\sum_{i=1}^{l} \frac{m_i}{H}} h^{\sum_{i=1}^{l} \frac{r_i}{H}} \right)^{H \cdot q} = g^{q \sum_{i=1}^{l} m_i} = (g^q)^{\sum_{i=1}^{l} m_i} = (g \wedge)^{\sum_{i=1}^{l} m_i} \tag{8}$$

We can use pollard's lambda method to solve out the aggregate value $\sum_{i=1}^{l} m_i$.

## 6. System Characteristic Analyses

In this section, we prove that the $(k, n)$-PDA has achieved the design goals including confidentiality, privacy preservation, fault tolerance, dynamic membership, and forward security.

*6.1. Confidentiality.* If attackers eavesdrop on the communication channel between entities, they may be able to obtain messages transmitted in the channels. But even if intercepting all the information, i.e., $(x_i, \sigma_{i\mathrm{AG}},\ C_i,\ \sigma_{i\mathrm{DSC}},\ C_{i\mathrm{DSC}})$, sent by all the SMs, the attackers cannot figure out the private data of any SM. Because if the attackers want to find privacy information from the ciphertext $C_i$ issued by $\mathrm{SM}_i$, they need $H1$ $(r_t|t|s_g) \bullet q$ to decrypt $C_i$. There are two ways to solve out $H$ $1(r_t|t|s_g) \bullet q$. One way is to calculate it with $r_t, s_g, q$, but $s_g$ is embedded in the SM, any illegal reading will be perceived, and $q$ is the secret key owned by the offline *OC* which adversaries cannot break. Another way is to collect no less than $k$ users' secret shares for reconstruction. However, before the reconstruction, attackers must decrypt $C_{i\mathrm{DSC}}$ to obtain $H1(r_t|t|s_g) \cdot y_i$ with DSC's private key $\mathrm{sk}_{\mathrm{DSC}}$ or collude with at least *k* SM*s*. As can be seen from the foregoing description, even if the attacker obtains data sent by all users and colludes with some (less than *k*) SMs, the decryption key cannot be reconstructed.

*6.2. Privacy Preservation.* Our solution aims to achieve data aggregation while protecting user data privacy. Although both AG and DSC are authorized users of the system, they can legally accept the data sent by users and complete the aggregation protocol, but they still cannot obtain users' fine-grained electricity consumption data. The following describes in detail that this solution can satisfy privacy requirement.

In our scheme, since SMs are honest and only receive parameters from the aggregator, SMs have no way to find any secret data of other users.

Although AG collects $(x_i, \sigma_{i\mathrm{AG}}, C_i)$ sent by each $\mathrm{SM}_i$, it cannot decrypt $C_i$ even owning $r_t$, because AG has no shares of the decryption key $H1(r_t|t|s_g) \cdot q$ to recover the key and also cannot calculate the key directly without $s_g$ and $q$.

Also, *DSC* can only obtain the aggregate ciphertext *C* from *AG*, and cannot obtain the ciphertext $C_i$ sent by a single user, so that even *DSC* can reconstruct the key $H1(r_t|t|s_g) \cdot q$ but cannot reveal any individual user's real-time usage with the key.

*6.3. Fault Tolerance.* As described in our scheme, when *k* or more SM*s* can send information to AG and DSC, the aggregation process can be executed correctly to obtain the accurate aggregate data of online SMs. In detail, $l\,(l \geq k)$ working SMs send messages to AG and DSC. After receiving messages from SM*s*, AG aggregates $C_i$ to obtain aggregate ciphertext *C* and then sends *C* to DSC. Next, DSC recovery is the key with *l* secret shares received from *l* SM*s* to decrypt the ciphertext *C* received from AG to obtain the accurate aggregation result of the *l* working SM*s*. That is to say, the proposed scheme can tolerate the failure of $n - k$ or fewer SMs and achieve accurate aggregation without the need for special processing. Therefore, the $(k, n)$-PDA is fault-tolerant.

*6.4. Dynamic Membership.* In the actual application environment, users may join in or exit the grid. Therefore, the

Table 2: System features comparison with related fault-tolerant schemes.

| Scheme | Confidentiality | Privacy | Forward security | No need for online high-authority entity* | Dynamic membership | Accurate aggregation |
|---|---|---|---|---|---|---|
| (Acs and Castelluccia 2011) [12] | √ | √ | × | √ | × | × |
| *SM*ART-ER [13] | × | √ | × | √ | × | × |
| PDAFT [14] | √ | √ | × | × | √ | √ |
| (Guan and Si 2017) [15] | √ | √ | × | × | × | × |
| FESDA [16] | √ | √ | √ | × | × | √ |
| PPFA [17] | √ | √ | × | × | √ | × |
| (Wang et al. 2020) [25] | √ | √ | √ | √ | √ | √ |
| Our scheme | √ | √ | √ | √ | √ | √ |

*Including online trusted authority.

aggregation scheme needs to support the random entry or exit of users with low communication traffic and computational cost. In $(k, n)$-PDA scheme, when a new user wants to join in a smart grid, the user applies to OC. After receiving the application, OC reviews the user's qualifications to determine whether to approve the application. If the application is approved, then OC assigns the user a group $g'$ with group key $s_g{}'$, chooses a new $\mathrm{id}_{\mathrm{new}}$, and evaluates the corresponding $y_{\mathrm{new}} = f(x_{\mathrm{new}})$, then sends $(s_g, id_{\mathrm{new}}, y_{\mathrm{new}})$ to the user. At the same time, the number of users in the group is increased by one. If a user wants to exit from the grid, he only needs to unregister his ID from the system and reclaim his smart meter. The number of meters in the group is reduced by one. It can be seen that the joining of new users and the exit of old users do not need to do anything for other users, which is completely in line with the actual application scenarios.

*6.5. Forward Security.* The proposed scheme is forward-secure. In other words, if an adversary breaks the system in the time slot $t_i$ and obtains the secret key $H1(r_{t_i}|t_i|s_g)\bullet q$, the adversary can only solve users' private information at $t_i$, but it cannot obtain any previous information. Because even the adversary has $H1(r_{t_i}|t_i|s_g)\bullet q$, it cannot derive $s_g$ and $q$. Furthermore, the random number $r_t$ distributed by the aggregator changes with time and then $H1(r_t|t|s_g)\bullet q$ updates with $r_t$. Therefore, the secret key of the time slot $t_i$ just affects to ciphertext at time $t_i$.

*6.6. System Feature Comparison.* In Table 2, we compare our scheme with several related fault-tolerant schemes [12–17, 25] in terms of whether it achieves some important features like confidentiality, privacy, forward security, the demand for an online trusted third party, dynamic membership, and accurate aggregation. Comparing with those schemes, $(k, n)$-PDA not only satisfies the necessary security and privacy requirements but also achieves the efficient dynamic membership management and accurate aggregate values without online high-authority entities or online trusted entities.

## 7. Efficiency Evaluation

In this section, we evaluate the efficiency of $(k, n)$-PDA on the computation cost and communication overload and make a comparison with some fault-tolerant schemes (Guan and Si 2017) [15], FESDA [16] (Wang et al. 2020) [25]). For convenience, we assume there are $n = 1000$ SMs in the aggregation domain, and $l = 990$ SMs out of the domain are working normally which is over the aggregation threshold required in the scheme. Furthermore, to evaluate efficiency, we set the secure parameter $\tau = \kappa = 512$, then $|p| = |q| = 512$ bits, $|N| = 1024$ bits, and $|N^2| = 2048$ bits, set the RSA module as 1024 bits, the length of the big prime in Shamir's secrete scheme as $|P| = 128$ bits, set timestamps and signatures as 64 bits, and set ID as 32 bits.

Table 3: Symbols of execution time for related operations.

| Symbol | Definition | Time (ms) |
|---|---|---|
| $T_H$ | Time for a hash computing | 0.001 |
| $T_e$ | Time for a modular exponentiation operation | 0.799 |
| $T_m$ | Time for a modular multiplication | 0.002 |
| $T_A$ | Time for an addition operation | 0.001 |
| $T_b$ | Time for a bilinear pairing operation | 1.823 |

*7.1. Computation Cost.* Generally speaking, the service center/control center (DSC in our scheme) has sufficient computing and storage resources, and all of the algorithms in our scheme and other peering schemes are widely used. Therefore, we only evaluate the computational workload on the terminal and the aggregator. Also, according to the assumptions in Adversarial Model and Assumptions, we do not count the computational overhead of signatures and verifications. Table 3 defines some symbols of executing time of related operations which are executed based on the PBC and OpenSSL library in a PC with 64-bit Windows 10 operating system, Intel Xeon E3 @3.5GHz CPU, and 8 GB memory.

*Computations on $SM_i$*: in the encryption stage, $(k, n)$-PDA needs $\mathrm{T}_H + 2T_e + T_m$ to compute $C_i$ and a $T_e$ to encrypt $H \cdot y_i$. In subsequent stages, and $\mathrm{SM}_i$ does not need to do any calculations. Therefore, the computational cost on each SM is $\mathrm{T}_H + 3T_e + T_m \approx 2.400$ ms in each time slot. In Guan and Si [15], it takes $\mathrm{SM}_i 2\mathrm{T}_H + 3T_e + 3T_m \approx 2.405$ ms to calculate $C_i$, $H_1$, and $H_2$. In FESDA [16], $\mathrm{SM}_i$ uses $2\mathrm{T}_H + 2T_e + 2$

Table 4: Comparison of computation cost.

| Scheme | Computation cost on one terminal (ms) | Computation cost on an aggregator (ms) |
|---|---|---|
| (Guan and Si 2017) [15] | $2T_H + 3T_e + 3T_m \approx 1.604$ | $(n-l)((n-1)T_a + 2T_m + T_e) + (n-1)T_m + T_e + T_H \approx 20.818$ |
| FESDA [16] | $2T_H + 2T_e + 2T_m \approx 2.406$ | $4T_H + (l-1)T_m \approx 1.982$ |
| (Wang et al.) [25] | $3T_b + 4T_e + 8T_m + 4T_H \approx 8.685$ | $(l-1)T_m \approx 1.978$ |
| Our scheme | $T_H + 3T_e + T_m \approx 2.400$ | $(l-1)T_m \approx 1.978$ |

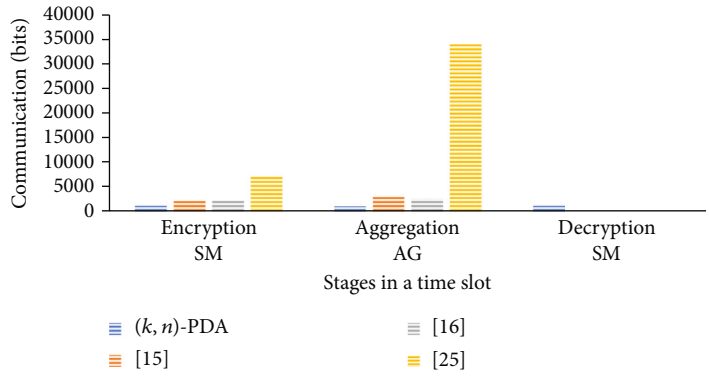


Figure 3: Comparison of real-time communication traffic.

$T_m \approx 1.604$ ms to get $C_i$ and $\text{MAC}_i$. In (Wang et al. [25], each user needs to update its blind factor through discussion with three or more users, so it takes at least $3T_b + 4T_e + 8T_m + 4T_H \approx 8.685$ ms to finish the encryption and blinding factor update. For the partners of a faulty SM, extra $4T_e + 2T_m + T_H \approx 3.201$ ms is needed to update the blind factor and reencryption. The more faulty users a user cooperates with, the more calculations are required. Therefore, compared with these peered schemes, the computational overload is acceptable for SMs in our scheme.

*Computations on the aggregator:* In the encryption stage, AG just generates a random number $r_t$. The computational workload of the generation of a random can be ignored. In the aggregation stage, AG needs $(l-1)T_m \approx 1.978$ ms to calculate $C = \prod_{i=1}^{l} C_i$. According to the descriptions in Guan and Si [15], the data aggregator (DA) needs $(n-l)((n-1)T_a + 2T_m + T_e)$ to deal with malfunctioning *SMs*. Then, DA spends $(n-1)T_m$ to compute $C_{\text{sum}}$ and $T_H + T_e$ to compute $\text{ENC}(\text{sk}_D, H_1)$. The amount of computational cost of DA in this stage is $(n-l)((n-1)T_a + 2T_m + T_e) + (n-1)T_m + T_e + T_H \approx 20.818$ ms. In FESDA [16], the fog nod (FN) spends $(l-1)T_m$ to aggregate received data to get $\widehat{C}$ and $4T_H$ to generate $\text{MAC}_j$ and $\text{MAC}_x$, so the computational cost in FN is $4T_H + (l-1)T_m \approx 1.982$ *ms*. In Wang et al. [25], the aggregator takes $(l-1)T_m \approx 1.978$ ms to aggregate the received ciphertext. According to the above comparison, the calculation amount on the aggregator in $(k, n)$-PDA is small.

The comparison of the calculation amount shows that our scheme is relatively friendly in terms of the calculation burden. The computation cost comparison among our scheme and others are illustrated in Table 4.

*7.2. Communication Cost.* The evaluation of communication can be considered due to real-time communication traffic demand. It reflects the real-time performance and the demand for communication capabilities of devices.

In the encryption phase of $(k, n)$-PDA, AG sends $r_t$ to each terminal. The traffic is $n * 512$ bits. Each SM sends $(x_i, \sigma_{i\text{AG}}, C_i)$ to AG that causes traffic of $(128 + 64 + 1024) = 1216$ bits. In contrast, each SM sends 2304 bits to DA in Guan and Si 2017, and each SM sends 2368 bits to FN in FESDA. In Wang et al. [25], normally, each user needs to send 2048 bits to the aggregator and 1024 bits to each cooperator. If some users fail to report their data, according to the description in [25], all users have to update their keys, recompute the ciphertext, and send the new ciphertext, which will cause another 2048 bits traffic for every SM. Therefore, with at least three partners, the amount of data each user needs to send is not less than 7168 bits. As we can see, in this stage, SMs in our scheme are easier with communication.

In the aggregation phase, AG sends $(\sigma_{\text{AG}}, C, \text{ID}_{\text{AG}})$ to DSC, which causes a communication overhead of $(64 + 1024 + 32) = 1120$ bits. Meanwhile, in Guan and Si [15], DA sends a message of 3072 bits to CC. In FESDA, if there is no malfunctioning SM, FN sends a message of 2304 bits to CC; otherwise, each malfunctioning SM will cause 32 bits traffic. In Wang et al. [25], the aggregator sends 2048 bits to the data center. If some failures occur, the aggregator needs to broadcast the identities of failed users to all users. That adds another $n \times 32 = 32000$ bits of communication overload. Also, in the aggregation process, the communication cost on AG in the proposed solution is lower than that in these peering schemes.

In the decryption phase of our scheme, DSC needs $k$ shares to reconstruct the key. So, before decryption, each working SM sends $(x_i, \sigma_{i\text{DSC}}, C_{i\text{DSC}})$ to DSC, generating 1216 bits of upload traffic. However, this traffic can be uploaded at idle time.

Figure 3 shows the comparison of communication traffic from a real-time perspective. The proposed solution distributes the communication volume to each stage in a more balanced manner, which allows more timely execution and lower bandwidth requirements.

## 8. Conclusion

This paper proposes the $(k, n)$-PDA scheme for smart grids to obtain the accurate aggregate real-time consuming data while ensuring users' privacy and achieving fault tolerance and dynamic membership without any online entity with high authorities. The analyses are present to prove that the proposed scheme meets all the design goals and system performance requirements.

## Data Availability

No data were used to support this study Yining Liu Guilin University of Electronic Technology, China.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

The study of the manuscript titled "Fault-tolerant Privacy-preserving Data Aggregation for Smart Grid" is funded by the National Natural Science Foundation of China under grant no. 61662016 and Key Projects of Guangxi Natural Science Foundation under grant no. 2018JJD170004.

## References

[1] H. Lam, G. Fung, and W. Lee, "A novel method to construct taxonomy electrical appliances based on load signaturesof," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 653–660, 2007.

[2] R. Anderson and S. Fuloria, "Who controls the off switch," in *2010 First IEEE International Conference on Smart Grid Communications*, pp. 96–101, Gaithersburg, MD, USA, 2010.

[3] S. Finster and I. Baumgart, "Privacy-aware smart metering: a survey," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 2, pp. 1732–1745, 2015.

[4] M. R. Asghar, G. Dan, D. Miorandi, and I. Chlamtac, "Smart meter data privacy: a survey," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2820–2835, 2017.

[5] P. Kumar, Y. Lin, G. Bai, A. Paverd, J. S. Dong, and A. Martin, "Smart grid metering networks: a survey on security, privacy and open research issues," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2886–2927, 2019.

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

[7] W. Jia, H. Zhu, Z. Cao, X. Dong, and C. Xiao, "Human-factor-aware privacy-preserving aggregation in smart grid," *IEEE Systems Journal*, vol. 8, no. 2, pp. 598–607, 2014.

[8] M. Badra and S. Zeadally, "Lightweight and efficient privacy-preserving data aggregation approach for the smart grid," *Ad Hoc Networks*, vol. 64, pp. 32–40, 2017.

[9] J. Song, Y. Liu, J. Shao, and C. Tang, "A dynamic membership data aggregation (DMDA) protocol for smart grid," *IEEE Systems Journal*, vol. 14, no. 1, pp. 900–908, 2020.

[10] Z. Sui and H. de Meer, "BAP: a batch and auditable privacy preservation scheme for demand response in smart grids," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 842–853, 2020.

[11] T.-H. H. Chan, E. Shi, and D. Song, "Privacy-preserving stream aggregation with fault tolerance," in *International Conference on Financial Cryptography and Data Security*, pp. 200–214, Springer, Berlin, Heidelberg, 2012.

[12] G. Ács and C. Castelluccia, "I have a DREAM!: differentially private smart metering," *IH'11 Proceedings of the 13th International Conference on Information Hiding*, pp. 118–132, 2011.

[13] S. Finster and I. Baumgart, "SMART-ER: peer-based privacy for smart metering," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS 2014) - INFOCOM Workshop on Communications and Control for Smart Energy Systems*, pp. 652–657, Toronto, ON, Canada, 2014.

[14] L. Chen, R. Lu, and Z. Cao, "PDAFT: a privacy-preserving data aggregation scheme with fault tolerance for smart grid communications," *Peer-to-Peer Networking and Applications*, vol. 8, no. 6, pp. 1122–1132, 2015.

[15] Z. Guan and G. Si, "Achieving privacy-preserving big data aggregation with fault tolerance in smart grid," *Digital Communications and Networks*, vol. 3, no. 4, pp. 242–249, 2017.

[16] A. Saleem, A. Khan, S. U. R. Malik et al., "FESDA: fog-enabled secure data aggregation in smart grid IoT network," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6132–6142, 2020.

[17] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: privacy preserving fog-enabled aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3733–3744, 2018.

[18] D. Abbasinezhad-Mood and M. Nikooghadam, "Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended Chebyshev chaotic maps," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4815–4828, 2018.

[19] D. Abbasinezhad-Mood and M. Nikooghadam, "An anonymous ECC-based self-certified key distribution scheme for the smart grid," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 7996–8004, 2018.

[20] D. Abbasinezhad-Mood and M. Nikooghadam, "Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications," *Future Generation Computer Systems*, vol. 84, pp. 47–57, 2018.

[21] J. Chen, G. Liu, and Y. Liu, "Lightweight privacy-preserving raw data publishing scheme," in *IEEE Transactions on Emerging Topics in Computing*, 2020.

[22] M. Jawurek, M. Johns, and K. Rieck, "Smart metering de-pseudonymization," *Proceedings of the 27th Annual Computer Security Applications Conference On*, pp. 227–236, 2011.

[23] S. Cleemput, M. A. Mustafa, E. Marin, and B. Preneel, "De-Pseudonymization of Smart Metering Data: Analysis and Countermeasures," *2018 Global Internet of Things Summit (GIoTS)*, pp. 1–6, 2018.

[24] K. Xue, B. Zhu, Q. Yang, D. S. L. Wei, and M. Guizani, "An efficient and robust data aggregation scheme without a trusted authority for smart grid," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1949–1959, 2020.

[25] X. Wang, Y. Liu, and K. R. Choo, "Fault tolerant, ulti-subset aggregation scheme for smart grid," in *IEEE Transactions on Industrial Informatics*, 2020.

[26] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," *TCC'05 Proceedings of the Second International Conference on Theory of Cryptography*, pp. 325–341, 2005.

[27] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, USA, 1996.

[28] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.