An Efficient Privacy-Preserving Data Aggregation Scheme without Trusted Authority in Smart Grid

Xinyu Zhao[1], Jinguo Li[1], Na Zhao[2], and Ping Meng[1]
(Corresponding author: Jinguo Li)
College of Computer Science and Technology, Shanghai University of Electric Power[1]
No. 1851, Hucheng Ring Road, Nanhui New Town, Pudong New Area, Shanghai, P.R. China
Email: lijg@shiep.edu.cn
Key Laboratory in Software Engineering of Yunnan Province, School of Software, Yunnan Kunming, P.R. China[2]
(Received Jan. 22, 2022; Revised and Accepted May 24, 2022; First Online July 3, 2022)

Abstract

To minimize communication overhead and safeguard individual data privacy, the aggregator in the intelligent grid aggregates the data collected by the user's smart meter (SM). In the field of smart grids, data aggregation has received considerable attention. However, most existing solutions rely on a trusted authority (TA) to distribute parameters, which is not always possible in real-world scenarios. Additionally, given the frequent changes in a user's membership, the system's efficiency should be increased. This paper proposes a data aggregation scheme for SMs that do not have a TA. Under these circumstances, this paper's solution can perform arbitrary ciphertext aggregation operations in response to the needs of data users while also supporting dynamic user management. The security analysis and simulation results demonstrate that this scheme can achieve the required security features while significantly reducing computational costs and communication overhead, making it more suitable for the next-generation smart grid.

Keywords: Arbitrary Aggregate Function; Data Aggregation; Dynamic User Management; Privacy-Preserving; Smart Grid

1 Introduction

A smart grid is a next-generation grid made up of existing power and communication systems. The power company can dynamically obtain the total power consumption and adjust the electricity price via command control and service query between the control center (CC) and users. Furthermore, it can provide additional power or recover energy based on user demand.

SM is a critical component of the smart grid that is installed on the user side. The CC receives real-time power consumption data from users via SMs. However, users of the smart grid cannot completely trust the cloud, and malicious users may obtain grid services illegally and even cause damage to the grid. Without adequate protection, grid services may be jeopardized, posing additional challenges: Individual privacy may be compromised as a result of real-time consumption data, from which an adversary can easily infer a user's life habits; (b) the computing power of SMs is limited, making some complex cryptographic operations impractical; and (c) dynamic user management becomes extremely complicated as a result of user membership changes or other unpredictable reasons [1]. As a result, a dependable, efficient, and secure mechanism for preserving privacy is critical for the smart grid.

Data aggregation has been used to protect users' privacy in a number of studies [3–21]. Aggregation of data is a frequent operation in IoT systems. Its primary objective is to efficiently aggregate and collect data with the goal of optimizing energy consumption, network life, traffic congestion, and data accuracy. The published data in the aggregation area has the same statistical characteristics as the original data. As a result, when we transmit the sum of the data, the individual data point is also masked. By utilizing a homomorphic encryption model, you can safeguard your privacy against the influence of data processors. Thus, the SM sends encrypted data to the local gateway, which aggregates it and uploads the aggregated encrypted data to the CC. Only CC has the ability to decrypt aggregated user data [2]. Lu *et al.* [3] and Abdallah *et al.* [5] aggregated users' power data using homomorphic encryption. However, the use of a public cryptographic system introduces significant computational overhead. To increase the efficiency of the system, a scheme [9, 10] was proposed to conceal the power data by masking the value. Typically, a TA generates and distributes the mask value. In [14, 15], a data aggregation scheme was used in the virtual aggregation area, but it was limited to performing

the aggregation function's operations.

This paper proposes a data aggregation scheme without a TA in the smart grid. The scheme is divided into four sections, with a collection of unique sequence numbers at its heart. This set enables the collection of data from all participants in order to perform any aggregation operations in the subsequent stage. At the same time, it ensures efficiency and computational security for a variety of thresholds. Our protocol makes the following contributions:

1) We safeguard users' privacy by assigning them unique numbers without the use of a TA, which is more scalable in practical applications.

2) According to the requirements, our proposed scheme enables arbitrary aggregation operations on ciphertexts.

3) To facilitate the management of dynamic users, we incorporate a threshold variable secret sharing scheme that is resistant to malicious user withdrawal and fault-tolerant.

The remainder of this paper is structured as follows. Section 2 discusses related work. Section 3 provides the system's model and threats. Then, in Section 4, we introduce some cryptographic preliminaries. Following that, Section 5 proposes our aggregation scheme, followed by its flexible user management. We demonstrate security analysis and efficiency evaluation in Section 6. Finally, Section 7 concludes this paper.

2 Related Work

In recent years, researchers have proposed various solutions to address the unique requirements of various application scenarios. To achieve secure data aggregation, some existing schemes employ homomorphic cryptosystems, while others employ one-time masked values, one-time padding, secure multi-party computation-based secret sharing, and various other techniques.

Since homomorphic encryption technology ensures that certain algebraic operations on plaintext can be performed directly on ciphertext, numerous aggregation schemes have been proposed using homomorphic encryption [10,13]. The authenticity of the message can be verified, and data aggregation can be accomplished with minimal communication and computational complexity using Lattice [1]. Users' consumption reports are aggregated at the gateway in Elgamal's study [15] to minimize communication overhead while effectively supporting fault tolerance for faulty smart meters. While the public key-based homomorphic encryption scheme is effective at protecting users' privacy, it imposes a significant communication burden on smart meters. Secure data aggregation is accomplished in schemes [3] and [18] using a symmetric cryptosystem and differential privacy. They also contribute to fault tolerance, but add to the cost as the number of faulty smart meters grows. Bao and Lu [3] proposed a lightweight data aggregation scheme that incorporates FT support and supports privacy, integrity, and differential privacy. Additionally, their solution works with any number of faulty smart meters. Shi *et al.* [18] employ a pairwise private stream data aggregation method that is both private and FT compatible. This scheme, however, is incapable of detecting a large number of faulty smart meters. Badra *et al.* [2] proposed a privacy-preserving data aggregation scheme that is both efficient and lightweight. However, in practice, it is difficult to perform full-trust authorization when a third-party agency distributes system parameters and transmits decryption keys [16].

Fog-enabled aggregation schemes have been investigated as a result of advancements in fog-based architectures for smart grids. Lu *et al.* [12] aggregates hybrid perception data in a fog computing environment using homomorphic Paillier encryption and the Chinese remainder theorem, and uses one-way hash chains to filter out fake data from unauthenticated devices early. These aggregate gation protocols, on the other hand, are limited to one or two simple statistical calculations, which are insufficient for performing various statistical analyses. Lyu *et al.* [14] proposed a fog-enabled data aggregation (PPFA) scheme in which nodes aggregate data from various smart grids on a periodic basis, and the cloud or provider aggregates the data from all fog nodes. To prevent data privacy leaks, the authors distribute noise among the parties using a Gaussian distribution. Wang *et al.* [21] proposed a secure aggregation scheme in a fog-based smart grid architecture based on anonymization. In this scheme, fog nodes aggregate data from sensor devices and forward these data to the cloud for long-term storage. However, this scheme aggregates data via a single fog device, which may be vulnerable to denial-of-service attacks and single point of failure. Additionally, this scheme's adversary model is limited, considering only possible insider attacks. Moreover, no additional fog nodes are integrated into the network in the event of a fog device failure to restore aggregated data.

As more smart meters gain Internet connectivity, aggregation areas overcome the limitations of traditional physical areas. Liu *et al.* [11] introduced the concept of a virtual aggregation area and proposes 3PDA, a practical privacy-preserving data aggregation scheme that does not require a third-party trusted authority. Among them, users who have a certain level of trust create a virtual aggregation area to conceal a single user's data. The disadvantage of their scheme is that certificate management is inefficient. Both [9] and [6] attempted to solve the user dynamic management problem through the use of homomorphic hashing and ID-based signatures, but neither was sufficiently secure. To preserve privacy, [5] distributed masks among users using a semi-honest model, but this scheme is relatively inefficient. Song *et al.* [20] proposed a scheme for managing dynamic users called dynamic membership data aggregation (DMDA). Xue *et*

al. [23] employed a super-increment sequence to encrypt multiple secrets into a single one with forward secrecy. Shen *et al.* [17] proposed a data aggregation scheme capable of producing accurate results in the face of malicious data mining attacks. Zuo *et al.* [24] proposed a privacy-preserving multi-dimensional data aggregation scheme for use in smart grids that does not rely on a third-party trusted authority. None of these, however, account for the possibility of malicious users systematically eavesdropping or destroying information. Xu *et al.* [22] managed user joining and leaving using a (t, n) threshold in secret sharing algorithm. When the initial threshold does not provide an adequate level of security, users must re-initialize and allocate secret shares, which is not only time consuming but may be impossible due to a lack of trusted communication channels.

In response to the above problems, this paper combined an improved secret sharing algorithm to propose a computationally secure solution without maintaining online dealers. The program has a small share size and a lower recovery complexity. Meanwhile, this paper optimized the encryption algorithm to further reduce computational complexity and enhance system efficiency.

3 System Model

3.1 Communication Model

The system model is designed according to [5, 8]. There are only two entity types involved: an untrusted aggregator center (AC) and n SMs $\{SM_i\}_{i=1}^{n}$, depicted in Figure 1.

SMs. SMs collect real-time usage data D_i, generate keys, and upload ciphertexts to the aggregator. All SMs in the system negotiate to generate shared keys. Generally, they are not assumed to be trusted; however, some of them with some trust relationships can cooperate when keys must be reconstructed. They can communicate with the aggregator through an unsecure bi-directional communication channel.

AC. AC is primarily responsible for initializing system parameters, as well as calculating and publishing aggregate function results. AC is typically assumed to be honest-but-curious, which means that it follows the protocol and does not tamper with the results of computations. However, it may collude with malicious participants in order to deduce some useful information. We assume that the aggregator conspires with no more than $n-2$ participants.

3.2 Design Goals

This paper presents a novel protocol that can compute arbitrary aggregation functions without requiring a TA, while supporting dynamic user management, reducing computing costs, and improving security level. Referring to [17–21], the design goals involve the following aspects:

Privacy: Forbidding sensitive user information disclosure and achieving $(n-k)$-source anonymity.

Eciency: Realizing low communication cost to be implemented to smart meters with limited computing power.

Scalability: Allowing a user to dynamically join in or quit from a smart grid system, while flexibly dealing with various thresholds.

4 Preliminaries

This section briefly introduces the homomorphic encryption, the Diffie-Hellman (DH) algorithm, and a TCSS scheme according to the Chinese Remainder Theorem (CRT).

4.1 Homomorphic Encryption

Homomorphic encryption provides a function for encrypted data processing. In this way, other people can process encrypted data, but original content will still not be revealed. Simultaneously, after the owner decrypts processed data, the result becomes exactly the processed result.

Let *Enc()* denote an encryption scheme, and K and d be its key and plaintext, respectively. F represents an operation. If for any instance *Enc()* of the encryption scheme and operation F, there exists an efficient algorithm Y such that:

$$Enc(K, F(d_1, ..., d_n)) = Y(K, F, (E(d_1), ..., E(d_n))) \quad (1)$$

We denote the encryption algorithm *Enc()* as homomorphic for operation F.

If $F(d_1, ..., d_n) = \sum_{i=1}^{n} d_i$, then the encryption scheme is an additively homomorphic encryption. If $F(d_1, ..., d_n) = \prod_{i=1}^{n} d_i$, then the encryption scheme is an multiplicatively homomorphic encryption. If Equation (1) holds for $F(d_1, ..., d_n)$ containing a mixed operation of addition and multiplication, then the encryption scheme is fully homomorphic [19].

4.2 Diffie-Hellman Algorithm

The DH algorithm [4] is one of the earliest key exchange algorithms. It enables both communicating parties to exchange keys securely in an insecure channel and encrypt subsequent communication messages.

In our protocol, we apply the DH algorithm key exchange algorithm to establish the shared keys. Let G denote a cyclic group with the prime order q, g be a generator of the group G and H be a hash function. The user utilizes the system parameters (G, g, q, H) to generate its

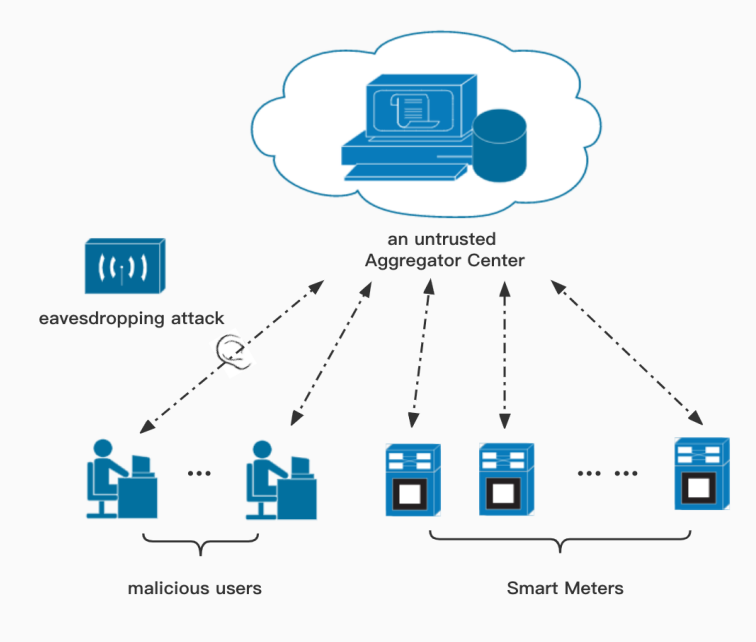

Figure 1: System model

public-private key pair (a, g^a), where a is the private key and g^a is the public key. This primitive is given as:

$$Agree(a_i, g^{a_j}) = S_{i,j} \tag{2}$$

where $S_{i,j} = H((g^{a_j})^{a_i})$.

4.3 The CRT-based $(t \rightarrow t', n)$ TCSS Scheme

In the CRT based TCSS scheme [7], threshold can be changed in an integer interval $[t, t']$ without updating the shares. A distinct threshold can be activated at any time through the public broadcast channel. The core of the scheme is to construct a novel matrix of primes P that utilizes a proposed sequence of nested closed intervals generated by large co-prime numbers. The dealer selects m sequences of pairwise co-prime integers that are arranged into the matrix form with the following order

$$\begin{aligned} &p_{1,1} < p_{2,1} < ... < p_{n,1} \\ &p_{1,2} < p_{2,2} < ... < p_{n,2} \\ &... \\ &p_{n,2} < p_{n,2} < ... < p_{n,n} \end{aligned} \tag{3}$$

where $m = t' - t + 1$. These integers need to satisfy

$$\prod_{i=1}^{t_l} \prod_{j=l}^{m} p_{i,j} > p_0 \prod_{i=1}^{t_l - 1} \prod_{j=l}^{m} p_{n-i+1,j} \tag{4}$$

$l = 1, 2, \ldots, m$.

We use the CRT based TCSS scheme to divide each user's private key into n parts. When required to regenerate the private key, the number of cooperating participants must reach t_l. The algorithm can be summarized as follows:

$$Share(K, t \rightarrow t', n) = \{K_i, i\}_{i \in [1,n]} \tag{5}$$

$$Recon(\{K_i, i\}_{i \in [1,n]}, t_l) = K \tag{6}$$

5 Proposed Aggregation Scheme

Our scheme will be described in three stages in this section: (a) initializing system parameters; (b) generating unique sequence numbers; and (c) performing aggregation operations. Then, we detail our scheme as follows. Table 1 summarizes the notations used in the paper.

Table 1: Frequently Used Notations

n	The number of SMs
m	The number of thresholds
k	The number of malicious SMs
M	The set $\{1, 2, ..., k\}$
SM_i	The i-th SM
f_e	A pseudo-random function indexed by e from f
Z_i	The random number selected by SM_i
y_i	The sample value of SM_i
$Seq(i)$	The unique sequence number of SM_i
T	The set of n SMs' sample values
D_i	The input data where the SM_i generates

5.1 The Initialization Phase

First, the AC initializes the system by generating the associated parameters for each participants and establishing flexible aggregation thresholds based on various circumstances, such as the likelihood of the SMs malfunctioning. The AC then uses the prime sequence generation algorithm in [7] to select $n \times m$ pairwise co-prime integers as (3) satisfying condition (4). Following completion of the preceding steps, the AC transmits the system parameters $(q, G, g, n, t \to t', P,$ and a constant $\alpha)$ to all SMs.

When participant SM_i receives the system parameter, it generates two pairs of secret keys $< C_i^{pk}, C_i^{sk} >$ and $< S_i^{pk}, S_i^{sk} >$. Applying the techniques shown in [4], SMs exchange their public keys (C_i^{pk}, S_i^{pk}) with each other to obtain their own shared key sets $\{S_{i,j}\}_{j\in[1,n],j\neq i}$ via the AC, where

$$\{S_{i,j}\}_{j\in[1,n],j\neq i} = \{Agree(S_i^{sk}, S_j^{pk})\}_{j\in[1,n],j\neq i} \tag{7}$$

We adopt the deterministic function f mentioned in [22], assuming there are $k(k \leq n-2)$ SMs that may collude with the AC. We let $M = \{1, 2, ..., k\}$. SM_i chooses a random number Z_i, and compute the following:

$$z_i = Z_i + \alpha \times p_0 \tag{8}$$

$$s_i^{sk} = S_i^{sk} + \alpha \times p_0 \tag{9}$$

After that, SM_i utilizes P to compute the shares separately on z_i and s_i^{sk} for various thresholds. The shares of SM_i for the threshold t_l are $\{z_{i,w}^{(l)}\}_{w\in[1,n]}$ and $\{s_{i,w}^{sk,(l)}\}_{w\in[1,n]}$. Then SM_i encrypts its information $(i, w, z_{i,w}^{(l)}, s_{i,w}^{sk,(l)})$ to obtain the ciphertext $\{e_{i,w}\}_{w\in[1,n]}$ and uploads it to the AC. Upon receiving all the ciphertexts $\{e_{i,w}\}_{w\in[1,n]} i \in [1, n]$ at the AC, it will share them with other SMs. During this period, SM_i uses Z_i along with $\{S_{i,j}\}_{j\in[1,n],j\neq i}$ to construct pseudo-random functions $PF_{i,w}(x)$ and $PF_i(x)$ as follows:

$$PF_{i,w}(x) = \begin{cases} f_{i,w}(x) \ mod \ M & i < w \\ -f_{i,w}(x) \ mod \ M & i > w \end{cases} \tag{10}$$

$$PF_i(x) = f_{Z_i}(x) \ mod \ M \tag{11}$$

where x represents the nonce information.

5.2 Unique Sequence Numbers Generation Phase

The proposed scheme considers the random sampling method and quicksort to establish a unique sequence number for each SM. The details are as follows:

1) SM_i randomly selects an integer sample value y_i from an interval $[1, n^{\alpha+2}]$, where $y_i \neq y_j$ for $\forall i \neq j (i, j \in [1, n])$, and uploads it to the AC;

2) Once the AC obtains all sample values $\{y_i\}_{i\in[1,n]}$, it applies quicksort to from small to large and obtains the sorted set T;

3) SM_i gets its own unique sequence number $Seq(i)$ based on the rank of its sample value in T.

5.3 Executing the Aggregation Operation Phase

In the reconstruction phase, the AC broadcasts the required threshold $t_l (l \in [1, m])$ and transmits a list of online SMs to each SM_i. Then the online SM_i presents its partial share $z_{i,w}^{(l)}$ to reconstruct z_w based on the CRT.

Each SM has an input data $D_i \in [1, n]^d$. According to $Seq(i)$, SM_i generates an n-dimensional vector $V_i = (v_i^1, \ldots, v_i^h, \ldots, v_i^n)$, where

$$v_i^h = \begin{cases} D_i + P_i(h) + \sum_{w\in[1,n]\setminus i} P_{i,w}(h) \ mod \ M \\ \qquad\qquad h = Seq(i) \\ 0 + P_i(h) + \sum_{w\in[1,n]\setminus i} P_{i,w}(h) \ mod \ M \\ \qquad\qquad h \neq Seq(i), h \in [n] \end{cases} \tag{12}$$

Afterward, SM_i sends V_i to the AC. Once the AC obtains $\{V_i\}_{i\in[n]}$, it performs the corresponding addition operation for each item in $\{V_i\}_{i\in[n]}$ to calculate $V_{agg} = (V_{agg}^1, \ldots, V_{agg}^j, \ldots, V_{agg}^n)$ as follows:

$$V_{agg}^j = \sum_{i=1}^{n} v_i^j \ mod \ M, j \in [n] \tag{13}$$

where V_{agg} consists of all SMs' information. Thus far, the AC can compute arbitrary aggregation functions using the vector V_{agg}.

5.4 Dynamic User Management

Our scheme has designed a dynamic user management mechanism with a detailed process demonstrated as follows.

1) SM Enrollment: We assume that a new SM (denoted as SM_r) is added to the system, which is similar to the steps in the initialization steps. First, SM_r generates two pairs of keys and sends the public keys (C_r^{pk}, S_r^{pk}) to the AC. Then SM_r negotiates the shared key set $\{S_{r,j}\}_{j\in[1,n'],j\neq r}$ and selects a random number Z_r. Since SM_i has received additional shared key from the new SM, SM_i updates its secret shared key set $\{S_{i,j}\}_{j\in[1,n'],j\neq i}$, where $S_{i,r}$ is the shared key that the new SM generates and shares with it. Afterward, $\{SM_i\}_{i\in[1,n']}$ shares the static secret (z_i, s_i^{sk}) to the AC using the CRT-based TCSS scheme and continues to perform the same operations as previously described.

2) SM Revocation: When a user SM_r is revoked from the system, it cannot upload its secret keys to the AC any more. Hence, AC should transmit a revocation message to the existing SMs. Consequently, SM_i sends its encrypted shares $\{z_{i,w}^{(l)}\}_{w\in[1,n']}$ and $\{s_{i,w}^{sk,(l)}\}_{w\in[1,n']}$ to the AC. AC can broadcast the

Table 2: Communication Overhead

Scheme	**CE-PPDA [5]**	**ICN-PPDA [22]**	**Our scheme**
Initialization	$O(n^2+2n)$	$O(3n^2+2n)$	$O((2m+1)n^2+2n)$
Sequence Number Generation	$O(n^2+n)$	$O(n^2+n)$	$O(n)$
Aggregation	$O(n^2)$	$O(n^2)$	$O(n^2)$

Table 3: The computation cost comparison

Scheme	**CE-PPDA [5]**	**ICN-PPDA [22]**	**Our scheme**
Sequence Number Generation	$O(n^2(\log n)^{4/3})$	$O(2n^3)$	$O(n\log n)$
Dynamic Joining Scheme	$O((n+r)^2(\log(n+r))^{4/3})$	$O(2(n+r)^3)$	$O((n+r)\log(n+r))$
Recovery Complexity	None	$O(t_l\log^2 t)$	$O(t_l)$

required threshold based on the situation to reconstruct the random number. Then SM_i continues to compute corresponding vector V_i and the AC will obtain the eventua aggregation result.

6 Performance Evaluation

In this section, we evaluate the performance of the proposed scheme in terms of security and complexity.

6.1 Security Analysis

The scheme attains privacy preservation and $(n-k)$ source anonymity under the proposed threat model discussed in Section 3.

We have assumed that there are k $(k \leq n-2)$ SMs that may collude with AC. Let $f_0 = \{f_1(\{D_1,\ldots,D_n\}),...,f_k(\{D_1,\ldots,D_n\}), f_{AC}(\{D_1,\ldots,D_n\})\}$, where $f_i(\{D_1,\ldots,D_n\})(i=1,...,k)$ and $f_{AC}(\{D_1,\ldots,D_n\})$ denote the outputs of SM_i and AC, respectively. If a polynomial time simulator S on the input $\{D_i\}_{i\in[1,k]}$ exists, then any probabilistic polynomial time protocol π calculates the function f privately, such that

$$S(\{SM_i\}_{i\in[1,k]}\cup AC, \{D_i\}_{i\in[1,k]}, f_0\}\overset{C}{\equiv} view_0^{\pi}(\{D_i\}_{i\in[1,n]})$$

where $\overset{C}{\equiv}$ denotes the computational indistinguishability. All SMs in the system will upload their information to the AC, which then deduces information of any honest user, implying that $\{view_i^{\pi}(\cdot): i\in[1,k]\}\subseteq view_{AC}^{\pi}(\cdot)$, where $view_i^{\pi}$ is the view of SM_i that executes the protocol π on the input.

The proof and lemma details are similar to those in [5] and [22], hence we need not discuss them here.

6.2 Efficiency Evaluation

In terms of storage cost, communication overhead, and computation cost, the proposed scheme is compared to other schemes with and without TA. This evaluation was conducted using a laptop equipped with an Intel Core i5-7267U processor running at 3.10GHz and 8 GB of RAM. Let m be the total number of changeable thresholds, n be the number of users of the initial system and r be the number of added users.

We compare our scheme's storage cost in the initialization phase to that of CE-PPDA [5] and ICN-PPDA [22]. All three schemes generate the user's shared key set using the DH algorithm. The DH algorithm generates keys with a length of between 512 and 1024, which must be a multiple of 64, with 1024 being the default. Therefore, the storage cost of each user's shared key set defaults to $1024(n-1)$ bits in ICN-PPDA [22] and our scheme, which is half of that in CE-PPDA [5]. This is because CE-PPDA [5] requires each user to generate two shared key sets.

The communication overhead of the aggregation phase is identical in CE-PPDA [5] and ICN-PPDA [22]. The first section contains costs associated with transmitting parameters $O(n)$, secret keys $O(n^2+n)$, shares $O(mn^2)$ and uploading ciphertexts $O(mn^2)$. The sharing threshold changes when the security environment's temporal dynamics are considered. As a result, our scheme has a higher computational overhead than the other two. We have enhanced the algorithm for generating unique sequence numbers in this article by increasing the number of variable thresholds. We save money by avoiding the transmission of n-dimensional vectors representing the number of sample values within the subinterval and cost $O(n^2)$. If r SMs join the system, there will be $O(rn^2)$ communication overhead reduced. Table 2 compares the actual communication overhead required for each stage.

We compare the computation cost of our scheme with CE-PPDA [5] and ICN-PPDA [22] in terms of unique sequence numbers generation phase and aggregation operation phase, depicted in Table 3. In CE-PPDA [5], the number of subintervals is $O(n^2(\log n)^{4/3})$ with a high probability, as proven by the author. To generate the sequence number for each participant in ICN-PPDA [22], participants generate the ciphertext of n-dimensional vectors, which has a computation cost of $O(n^3)$. Follow-

ing that, the aggregator executes algorithms to obtain the unique sequence number set, which in the worst-case scenario costs $O(n^3)$. Due to the dynamic nature of some participants joining and leaving the system, there is some additional overhead. Because the operations used to generate specific sequence numbers remain constant, the computational overhead is $(n+r)log(n+r)$. When some users log out of the system, the operation becomes almost the same as the basic scheme. Meanwhile, for different thresholds in the interval $[t, t']$, the (t, n) SS scheme must be used m times repeatedly to recover the random number Z_i. In comparison, our scheme has a smaller share size and a negligible loss of secret entropy when an insufficient number of participants attempt to tamper with the system in order to reveal the secret. The corresponding computation cost in this paper is simply the addition of the quick sort and recovery complexity.

Following that, we run simulation experiments to determine the computational costs of three different schemes and validate the final results. We increased the number of SMs (n) in the experiment in an even manner from two to twenty. Due to the requirement to allow users to dynamically exit and rejoin the system, ICN-PPDA [22] has a higher computation overhead than CE-PPDA [5]. Figure 2 illustrates the experimental results. As shown in the figure, our solution has a much lower computation overhead than the other two schemes in the second stage. Our scheme has demonstrated a high level of efficiency. This is more conducive to meeting the scenario requirements of SMs in real-world situations.

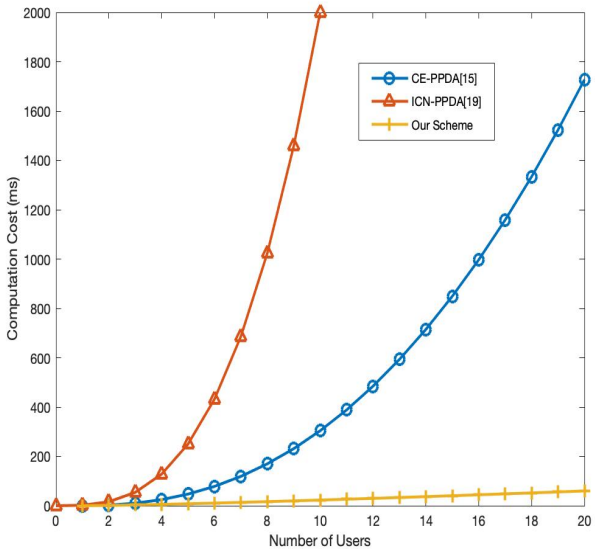

Figure 2: Computation Cost about Generating Sequence Number

7 Conclusion

We propose an efficient, privacy-preserving, and robust data aggregation scheme for the smart grid in this paper. There is no requirement for a trusted authority in our scheme, and computing arbitrary aggregation functions can be accomplished by generating unique numbers for SMs. At the same time, it supports dynamic user management and enhances security through the use of the TCSS algorithm. Our performance analysis demonstrates that our scheme significantly reduces communication overhead and computation costs, which meets the design objectives well.

Acknowledgments

This work has been supported by the National Key Research and Development Program No.2018YFB2100100, Postdoctoral Science Foundation of China No.2020M673312.

References

[1] A. Abdallah and X. S. Shen, "A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 396–405, 2016.

[2] H. Bao and R. Lu, "Lightweight and efficient privacy-preserving data aggregation approach for the smart grid," *Ad Hoc Networks*, vol. 64, pp. 32–40, 2017.

[3] H. Bao and R. Lu, "A lightweight data aggregation scheme achieving privacy preservation and data integrity with differential privacy and fault tolerance," *Peer-to-Peer Networking and Applications*, vol. 10, no. 1, pp. 106–121, 2017.

[4] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[5] X. Gong, Q.-S. Hua, L. Qian, D. Yu, and H. Jin, "Communication-efficient and privacy-preserving data aggregation without trusted authority," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 1250–1258, 2018.

[6] P. Gope and B. Sikdar, "An efficient data aggregation scheme for privacy-friendly dynamic pricing-based billing and demand-response management in smart grids," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3126–3135, 2018.

[7] X. Jia, D. Wang, D. Nie, X. Luo, and J. Z. Sun, "A new threshold changeable secret sharing scheme based on the chinese remainder theorem," *Information Sciences*, vol. 473, pp. 13–30, 2019.

[8] T. Jung, X. Mao, X.-Y. Li, S.-J. Tang, W. Gong, and L. Zhang, "Privacy-preserving data aggregation without secure channel: Multivariate polynomial evaluation," in *2013 Proceedings IEEE INFOCOM*, pp. 2634–2642, 2013.

[9] F. Knirsch, G. Eibl, and D. Engel, "Error-resilient masking approaches for privacy preserving data aggregation," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3351–3361, 2016.

[10] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "Eppdr: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2053–2064, 2013.

[11] Y. Liu, W. Guo, C.-I. Fan, L. Chang, and C. Cheng, "A practical privacy-preserving data aggregation (3pda) scheme for smart grid," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1767–1774, 2018.

[12] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[13] 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.

[14] 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.

[15] J. Ni, K. Zhang, K. Alharbi, X. Lin, N. Zhang, and X. S. Shen, "Differentially private smart metering with fault tolerance and range-based filtering," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2483–2493, 2017.

[16] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Edat: Efficient data aggregation without ttp for privacy-assured smart metering," in *2016 IEEE International Conference on Communications*, pp. 1–6, 2016.

[17] H. Shen, Y. Liu, Z. Xia, and M. Zhang, "An efficient aggregation scheme resisting on malicious data mining attacks for smart grid," *Information Sciences*, vol. 526, pp. 289–300, 2020.

[18] Z. Shi, R. Sun, R. Lu, L. Chen, J. Chen, and X. S. Shen, "Diverse grouping-based aggregation protocol with error detection for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2856–2868, 2015.

[19] L. Shundong, D. Jiawei, and W. Daoshun, "Survey on homomorphic encryption and its applications to cloud security," *Journal of Computer Research and Development*, vol. 52, no. 6, p. 1378, 2015.

[20] 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, 2019.

[21] H. Wang, Z. Wang, and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 712–719, 2018.

[22] C. Xu, L. Zhang, L. Zhu, C. Zhang, X. Du, M. Guizani, and K. Sharif, "Aggregate in my way: Privacy-preserving data aggregation without trusted authority in icn," *Future Generation Computer Systems*, vol. 111, pp. 107–116, 2020.

[23] K. Xue, B. Zhu, Q. Yang, D. S. 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, 2019.

[24] X. Zuo, L. Li, H. Peng, S. Luo, and Y. Yang, "Privacy-preserving multidimensional data aggregation scheme without trusted authority in smart grid," *IEEE Systems Journal*, 2020.

Biography

Xinyu Zhao is currently studying for a master's degree in computer technology from the School of Computer Science and Technology, Shanghai Electric Power University. Research direction: network security, cloud computing, smart grid, etc. E-mail: zhaoxy@mail.shiep.edu.cn.

Jinguo Li (IEEE M'16), received a bachelor's degree in information security from Hunan University in 2007 and a doctorate degree in computer science and technology from Hunan University in 2014. He is currently an associate professor at the School of Computer Science and Technology, Shanghai Electric Power University. His research activities focus on information security and privacy, applied cryptography and smart grids.

Na Zhao received the Ph.D. degree from the Yunnan University in 2011. She is a Associate Professor with the School of Software, Yunnan University. Her research interests include software engineering, complex network, the Internet of things, smart grid, etc.

Ping Meng is currently studying for a master's degree in computer technology from the School of Computer Science and Technology, Shanghai Electric Power University. Her research area is computer vision. E-mail: mengping@mail.shiep.edu.cn.