An Efficient and Robust Data Aggregation Scheme Without a Trusted Authority for Smart Grid

Kaiping Xue, *Senior Member, IEEE*, Bin Zhu, Qingyou Yang, David S. L. Wei, *Senior Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—Secure data aggregation has been widely studied in the area of the smart grid. Many existing schemes have studied protecting user's privacy in data aggregation by using advanced cryptographic tools. However, they usually introduce a large computation burden to smart meters in limited computing power or require a trusted authority. How to ensure the efficiency on the user side while preserving user's privacy still has not been well addressed. In this article, we consider the scenario where there does not exist a trusted authority and users in the smart grid may dynamically change, and propose an efficient and robust data aggregation scheme without a trusted authority for the smart grid. Our proposed scheme not only ensures user's privacy and efficiency but also supports flexible dynamic user management with no need of involving a trusted authority. Analysis of security and performance shows that our scheme can guarantee stronger security features, while ensuring efficiency in terms of computation, communication, and storage overhead.**

***Index Terms*—Data aggregation, dynamic user management, privacy preserving, smart grid.**

I. Introduction

SMART grid has been considered as the next-generation power supply system, which incorporates the advanced information and communication technology (ICT) and facilitates a two-way communication between users and power utility [1], [2]. With the use of communication networks, the smart grid can collect and analyze the status of power generation, transmission, and consumption, and respond to the changes of load demand in real time. In a smart grid, the smart meter is one of the most important components installed at the user side. It facilitates residential users to report their real-time power usage data (e.g., every 15 min) to the control center (CC). Based on the reported data, the CC can monitor and predict the power consumption, allocate and balance loads and resources, and adjust the power generation and dynamic electricity prices according to the demand, and so on.

Although real-time data collection can significantly improve the quality of services (QoSs) provided by the utility, this process also poses some security and efficiency challenges. On the one hand, users' privacy may be leaked out from their real-time consumption data. In fact, it is easy for an adversary to infer a user's life habits from its real-time consumption data, such as when a user leaves his/her house [3], [4]. Therefore, some cryptographic algorithms should be implemented to preserve users' privacy. On the other hand, usually smart meter has limited computing power, which is impractical to execute some complex cryptographic operations. Moreover, dynamic user management is also an important concern. User migration and utility company changing will lead to user's membership changes. In [5] and [6], the concept of the virtual aggregation area is mentioned. User membership also changes frequently in the virtual aggregation area. Therefore, the overhead of user enrollment and user revocation need to be reduced. Furthermore, in an aggregation area with a large number of users, users may often malfunction because of the limited lifetime of smart meters or natural disasters, and they may not even be able to inform other entities about their malfunctions. This indeed makes the user management extremely complicated and challenging.

Recently, researchers have paid much attention to privacy-preserving data aggregation in the smart grid. Lu *et al.* [7], Abdallah and Shen [8], Liu *et al.* [5], and Zhu *et al.* [9] used the public key cryptosystem of additional homomorphic encryption to aggregate users' power data, respectively. In recent years, we also proposed some solutions to deal with specific requirements in different application scenarios, e.g., [10]–[13]. Users' privacy can be preserved in these schemes, but a large computation overhead is also introduced due to leveraging the public cryptosystem. To make the proposed data gathering and aggregation protocol more efficient while still preserving user privacy, some masking-based schemes [6], [14]–[16] have been proposed. These schemes usually utilize a simple homomorphic encryption which is not based on the public cryptosystem, but through a masking value to obfuscate the power data. The masking value is usually generated and distributed by a trusted authority [14], [16], [17]. Meanwhile, Gong *et al.* [15] proposed a distribution method to generate the masking value when there is no trusted authority. However, without a trusted authority, dynamic user

Manuscript received August 10, 2019; revised November 15, 2019; accepted December 16, 2019. Date of publication December 24, 2019; date of current version March 12, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61972371, and in part by the Youth Innovation Promotion Association of the Chinese Academy of Sciences under Grant 2016394. *(Corresponding author: Kaiping Xue.)*

Kaiping Xue and Bin Zhu are with the Department of Electronic Engineering and Information Science and the School of Cyber Security, University of Science and Technology of China, Hefei 230027, China (e-mail: kpxue@ustc.edu.cn).

Qingyou Yang is with the Baidu International Technology (Shenzhen) Company Ltd., Shenzhen 518000, China.

David S. L. Wei is with the Computer and Information Science Department, Fordham University, New York, NY 10458 USA.

Mohsen Guizani is with the Department of Computer Science and Engineering, Qatar University, Doha, Qatar.

Digital Object Identifier 10.1109/JIOT.2019.2961966

management becomes more complicated. Song *et al.* [6] utilized a dynamic membership data aggregation scheme for the virtual aggregation area in the smart grid. However, it is not suitable to deal with the problem of user malfunction.

Motivated by this observation, we conduct this article to provide an efficient and robust data aggregation scheme without a need of involving trusted authority for the smart grid. In our scheme, to achieve user privacy preserving during the whole process, we do not introduce complicated cryptographic computation to the system, so that our scheme can still be implemented by smart meters in limited capability. Moreover, the dynamic user management is also supported with no need of a trusted authority involved in the system. To sum up, this article makes the following three main contributions.

1) We propose a computation-efficient and privacy-preserving data aggregation scheme which also includes flexible dynamic user management with no need of any trusted authority. An efficient homomorphic encryption is used to avoid vast computation and the (t, n) threshold secret sharing scheme is introduced to enable flexible dynamic user management.
2) We extend Shamir's (t, n) secret sharing scheme to support dynamic secret sharing, in which a secret can change frequently while its shares do not need to update synchronously, thereby ensuring the efficiency of our protocol.
3) A well-designed key negotiation and sharing mechanism is proposed, which can improve the security level of our scheme to N-source anonymity, while ensuring the system's robustness.

The remainder of this article is organized as follows: Section II discusses the related work. We review the basic blocks for our scheme as preliminaries in Section III. In Section IV, we introduce our network model, security model, and design goals. Then, we detail our proposed scheme in Section V, followed by security analysis and performance analysis in Sections VI and VII, respectively. Finally, Section VIII concludes this article.

II. Related Work

The security and privacy issue is one of the main obstacles for the development of the smart grid, which has been identified in the literature, e.g., [3], [18], [19], and [11]. Generic security protection methods in traditional network environments, such as source authentication and integrity protection mechanisms, have thus been introduced to construct a more secure communication between different components in the smart grid. However, owing to the distinctive features of the smart grid, users' privacy may still be leaked out unconsciously, which will further result in a more serious loss [20], [21].

With the increase in privacy concerns, some considerable privacy-preserving schemes have been proposed for the smart grid. One of the most important categories is secure data aggregation, in which the CC is only allowed to access the aggregated results but not the user's individual data, thus ensuring the privacy of users' information. To achieve a privacy-preserving data aggregation, homomorphic encryption, which allows computing on ciphertext, is usually utilized in academic researches. In 2012, Lu *et al.* [7] used Paillier's homomorphic encryption [22] and proposed a novel efficient and privacy-preserving aggregation scheme (EPPA) to process the data of all dimensions as a whole rather than do it separately. After that, some other Paillier-based data aggregation schemes are proposed to solve privacy issues of the smart grid in different aspects. Li *et al.* [23] proposed an efficient privacy-preserving demand response (EPPDR) scheme which realizes privacy-preserving electricity demand aggregation and efficient response. In [12], we proposed a privacy-preserving multisubset data aggregation, named PPMA, and then in [13], we proposed a novel scheme, named PPSO, to solve the privacy issues in service outsourcing. Meanwhile, Boneh–Goh–Nissim (BGN) homomorphic encryption [24] is also a practical technology used recently [25], [26], which is more efficient than Paillier especially when the plaintext space is very small. Besides, some other public-key homomorphic encryption schemes, such as lattice based [8] and Elgamal based [27] have also been proposed to achieve privacy preservation for data aggregation in the smart grid. Liu *et al.* [5] proposed a privacy-preserving data aggregation scheme without TTP based on the lifted EC-ElGamal cryptosystem, named 3PDA.

Although users' privacy can be well protected by utilizing the public-key-based homomorphic encryption schemes, large computation burden is also brought in to smart meters in limited computing power. To make the protocol efficient and practical, some schemes based on symmetrical homomorphic encryption (also called the masking approach) [28] have been proposed. In these schemes, a masking value is utilized to obfuscate user's metering data, and it is infeasible to access user's real data without knowing its masking value. In most masking-based schemes (e.g., [14], [16], [17], [29], and [30]), masking value is distributed by a trusted authority, which also delivers the corresponding decryption key to CC and helps manage user enrollment and revocation. However, a totally trusted authority is hard to conduct in practice. According to this consideration, some masking approaches of data aggregation without any trusted authority have been proposed [6], [15], [31]. Without any trusted authority, masking values can be negotiated among users in a distribution way as Gong *et al.* [15] do, but the dynamic user management is still difficult and has not been addressed adequately in their work. When a user registers to or quits from the system, all users need to renegotiate their masking values, which will significantly bring in large communication overhead. Although Knirsch *et al.* [31] use homomorphic hash to address the problem of dynamic user in masking approaches, some shortages in [31] on security have been identified in [14]. Since we cannot ensure the security of the hashed data when the plaintext space is small, an attacker can compute the original message by traversing the plaintext space. Song *et al.* [6] utilized homomorphic encryption and ID-based signature to address the user dynamic problem in virtual aggregation area. Users who want to exit have to participant in the logout phase, but a malfunctioning user may not be able to send messages to other entities. Therefore, this scheme is not suitable to deal with user malfunctions.

Therefore, in this article, we aim to propose a masking-based data aggregation scheme, which can keep efficiency while enforcing a higher security guarantee. Moreover, our scheme supports flexible dynamic user management even without a trusted authority.

III. Preliminaries

The basic building blocks of our scheme are the homomorphic encryption, the $(t,\ n)$ threshold secret sharing scheme, and the superincreasing sequence, which are briefly introduced in the following sections.

A. Homomorphic Encryption

Homomorphic encryption is a set of secure algorithms which allow certain algebraic operations on the plaintext to be performed directly on the ciphertext. Generally, assume a homomorphic encryption algorithm $Enc(\cdot)$, and two messages m_1 and m_2, with which we can directly compute $Enc_k(m_1 \odot m_2) = Enc_{k_1}(m_1) \circ Enc_{k_2}(m_2)$ on the ciphertexts without learning the plaintext m_1, m_2, and the security key. In fact, $\odot$ represents addition or multiplication operation, correspondingly called as additional homomorphic encryption and multiplication homomorphic encryption. It should be noted that for the scenario of data aggregation in a smart grid, our design only requires an efficient additional homomorphic encryption in order to avoid high complexity computation. More specifically, we adopt an efficient additional homomorphic cryptosystem, proposed in [28], which is mainly composed of the following steps.

1) *Encryption:* To encrypt a message $m \in [0, M-1]$, where M is the modulus, the entity chooses a random security key $sk \in [0, M-1]$ and computes the ciphertext c as

$$c = Enc_{sk}(m) = m + sk \mod M. \tag{1}$$

2) *Decryption:* Given the ciphertext c and its security key sk, the plaintext can be obtained by executing

$$m = Dec_{sk}(c) = c - sk \mod M. \tag{2}$$

3) *Additional Homomorphism Property:* Let $c_1 = Enc_{sk_1}(m_1)$ and $c_2 = Enc_{sk_2}(m_2)$, we can obtain the ciphertexts of $m_1 + m_2$ by aggregating ciphertexts as

$$c_A = c_1 + c_2 \mod M = Enc_{sk_A}(m_1 + m_2) \tag{3}$$

where $sk_A = sk_1 + sk_2 \mod M$. It should be noted that to support the homomorphic operation, the chosen module M must be greater than the aggregated messages, that is, if N ciphertexts are added, M must be larger than $\sum_{i=1}^{N} m_i$. In practice, if $l = \max_i\{m_i\}$, M must be chosen as $M = 2^{\lceil \log_2(l*N) \rceil}$.

B. (t, n) Threshold Secret Sharing With Dynamic Secret

A *(t, n)* threshold secret sharing scheme is a method for sharing a secret among a group of n participants in such a way that any t or more participants can reconstruct the secret, but no set of $t-1$ or fewer can do so.

1) Shamir's Threshold Scheme: The most basic threshold secret sharing scheme is proposed by Shamir [32], which is introduced as follows.

a) Initialization: Let the group of participants be denoted by $P = P_1, P_2, \ldots, P_n$, and the party who wants to share its secret is called as *dealer*. The dealer chooses n distinct nonzero elements from $\mathbb{Z}_p$, denoted x_i, $1 \le i \le n$, where $p \ge n+1$ is a prime number.

b) Share distribution: Let $S \in \mathbb{Z}_p$ be the secret that the dealer wants to share. First, the dealer randomly chooses $t-1$ elements $a_1, a_2, \ldots, a_{t-1} \in \mathbb{Z}_p$, and let $a_0 = S$. Then, the dealer constructs a polynomial of degree $t-1$ as

$$f(x) = \sum_{j=0}^{t-1} a_j x^j \mod p \tag{4}$$

and computes the shares $y_i = f(x_i)$ for $1 \le i \le n$. Finally, the dealer securely distributes the share (x_i, y_i) to participant P_i for $1 \le i \le n$.

c) Secret reconstruction: Any party who collects t or more different shares can reconstruct secret S. First, the party reconstructs the polynomial using t shares by computing

$$f(x) = \sum_{i=0}^{t} f(x_i) \prod_{j=0, j\neq i}^{t} \frac{x - x_j}{x_i - x_j} \mod p. \tag{5}$$

Then, the secret can be computed as $S = f(0)$.

2) Extend to Support Dynamic Secret: In this article, the secret will change frequently because of user revocation and registration. Therefore, the basic Shamir's threshold scheme is not suitable for this article, since it must update all the shares every time when the secret S changes, which brings in significant communication overhead. Thus, we further extend the basic Shamir's threshold scheme to support efficient dynamic secret sharing.

To share a secret $S \in \mathbb{Z}_p$ which may change frequently, the dealer first randomly chooses another secret ss (called *static secret*), and shares the static secret ss to n participants by Shamir's threshold scheme as the above mentioned steps. Then, the dealer chooses a random number $z \in \mathbb{Z}_p$, computes T_z using (6), and publishes the parameter pair (z, T_z)

$$T_z = S - H(ss \oplus z) \mod p \tag{6}$$

where $H(\cdot)$ is a secure one way function, and $\oplus$ is the bit-wise exclusive-OR operation.

To reconstruct the secret S, any party first reconstructs the static secret ss using Shamir's threshold scheme, then obtain secret S by computing

$$S = T_z + H(ss \oplus z) \mod p. \tag{7}$$

When updating the secret from S to S', the dealer only needs to choose a new random number $z' \in \mathbb{Z}_p$, computes a new pairs

$(z', T_{z'})$ as (6), and publishes it. Then, the secret S' can also be reconstructed by computing (7). In this process, the share of each participant [e.g., (x_i, y_i)] does not need to generate and distribute again, and thus the communication overhead can be largely reduced.

C. Superincreasing Sequence

In mathematics, a sequence of positive real numbers $b_1, b_2, \ldots,$ is called *superincreasing* if every element of the sequence is greater than the sum of all the previous elements in the sequence. Formally, they can be written as

$$b_{n+1} > \sum_{i=1}^{n} b_i. \tag{8}$$

In this article, we use a superincreasing sequence to encode multiple secrets into one secret. Generally, assume the dealer has k secrets $s_1, s_2, \ldots, s_k \in \mathbb{Z}_p$ needed to be shared. The dealer first generates superincreasing sequence $b_1, b_2, \ldots, b_k$ such that $b_1 = 1$, $b_j > \sum_{i=1}^{j-1} s_i b_i$, then the k secrets can be encoded as

$$S = b_1 s_1 + b_2 s_2 + \cdots + b_k s_k. \tag{9}$$

To decode the secret S, one can obtain $s_1, s_2, \ldots, s_k$ by iteratively computing

$$s_i = \frac{S - (S \mod b_i)}{b_i}, \quad S = S - s_i \cdot b_i \tag{10}$$

for $i = k$ to 1.

IV. System Model

In this section, we first describe the network model and security model of our proposed scheme. Then, the design goals for our scheme are also presented.

A. Network Model

As shown in Fig. 1, we consider a hierarchical structure of the smart grid communications, where a CC connects with multiple building area networks (BANs). Each BAN is formed by a group of users equipped with smart meters that connect to a BAN gateway. The following presents the details of each component.

1) *Smart Meter/User:* Smart meters are intelligent devices that are installed at user premises by utility companies. They measure the power usage and report the measured results to CC periodically. For clarity, we can easily treat a smart meter and the associated user as the same component.
2) *BAN/Gateway:* A BAN is a group of users usually living in the same building. Generally, a BAN via a gateway connects to the CC via a wired connection. In fact, the BAN gateways are usually deployed by a third-party operator (e.g., telecom operator). They serve the users and the CC, and forward the communication message between the users and CC.
3) *CC:* The CC is responsible for collecting, processing, and analyzing the data reported from users. With this information, the CC can adjust the electricity generation, transmission, and distribution to meet the dynamic demands. Additionally, the CC also takes the responsibility for the system initialization and management of dynamic users.

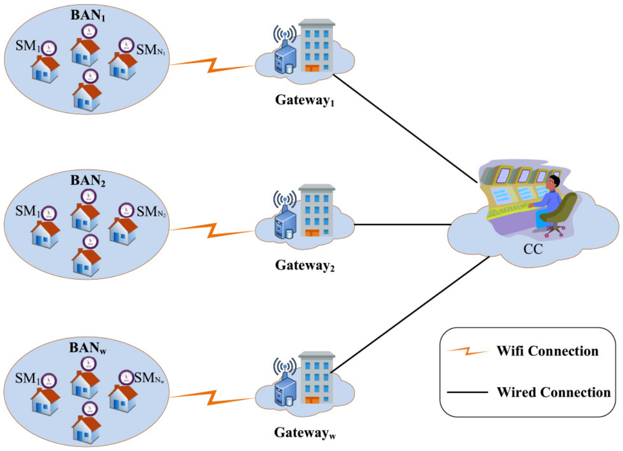


Fig. 1. Network model.

B. Security Model

In this article, we adopt the semihonest security model based on which all participants (including CC and BAN gateways) are assumed to be curious but honest. Generally, users will follow the defined protocols to interact with other parties exactly and do not tamper the results of the computations. Our scheme also assumes that a user can securely establish a shared key with another user, which can be implemented by the Diffie–Hellman key exchange protocol [33]. The CC will faithfully execute the protocol and will not cheat users that may damage the reputation of the utility. However, all entities are curious about other users' private information, and they may eavesdrop the communication channels and try to infer other user's individual data through listening to the traffic and the message. Moreover, we assume there are at most $\min\{t, k + k'\}$ users that can collude (the meaning of t, k, and k' are illustrated in Table I). Otherwise, they can deduce other users' private information in our model. The collusion between the CC and the gateway is not considered, as it may damage the reputation of the utility companies. This article mainly focuses on the privacy and confidentiality of users' data that are transmitted between the users and the CC. Other security properties, such as authentication and integrity are also important but beyond the scope of this article. The latter has been achieved in [6], [14], [16], and [34].

C. Design Goals

We intend to design an efficient and robust data aggregation scheme for the smart grid that can guarantee users' privacy preserving and meanwhile also keeps the system's efficiency. Specifically, our design goals comprise the following aspects.

1) *Efficiency on User Side:* The designed protocol should be efficient on the user side so that it could be implemented to the smart meters in limited computing power.

TABLE I
NOTATIONS OF OUR SCHEME

Notations	Descriptions
About Homomorphic Encryption	
m_{ij}, c_{ij}	Metering/Encrypted data of U_i at time slot j
M	Encryption modulus
k_{ij}	The key for U_i to share with U_j
$\mathbb{K}_{ij}$	U_i's shared key set at time slot j
k, k', K	U_i shares k keys with others while receiving k' keys, K is the defined upper bound of $\lvert\mathbb{K}_{ij}\rvert = k+k' \leq K$
sk_{ij}	Encryption key computing from $\mathbb{K}_{ij}$
About $(t,\ n)$ Threshold Secret Sharing Scheme	
p	A large prime number
ss_i	Static secret of U_i
S_{ij}	Dynamic secret of U_i at time slot j
$(t,\ n)$	n shares, at least t shares can be used to recover the secret
$\{a_{1,i}, a_{2,i}, ..., a_{t-1,i}\}$	Parameters of $(t,\ n)$ threshold secret sharing scheme which are chosen by U_i
$(x_j, f(x_j))$	Share for U_j
$(z_{ij}, T_{z_{ij}})$	Parameter pair to recover the dynamic secret S_{ij}
Others	
N	Number of users in aggregation result
$\mathbb{U}, \mathbb{U}^i_{neg}, \mathbb{U}^i_{sha}$	The user set of whole system, key negotiation, secret sharing
$H_1(\cdot), H_2(\cdot)$	Secure one-way hash function

2) *Data Confidentiality:* The proposed scheme should achieve the confidentiality of individual users' real-time energy consumption data, even if an adversary eavesdrops the communication channel or compromise several vulnerable users.
3) *Privacy Preservation:* No one, including the CC, is allowed to learn users' privacy information from real-time energy consumption data.
4) *Flexible Dynamic User Management:* The proposed scheme should provide a flexible users enrollment and revocation mechanism for supporting a user to dynamically join in/quit from a smart grid system. Moreover, when some smart meters are malfunctioning and fail to submit real-time consumption data, the CC should still be able to decrypt the ciphertexts to get correct results from the correct functioning meters.
5) *N-Source Anonymity:* For each user included in an aggregated result, there should be at least $N-1$ other users whose data are also contained in the result and are indistinguishable [35].
6) *Forward Secrecy:* It requires that the disclosure of the current security key would not affect the confidentiality of the previous individual data.

V. Proposed Scheme

In this section, we first give the overview of our proposed scheme. Then, the detailed phases are also presented.

A. Overview

The purpose of our proposed scheme is to construct an efficient and robust data aggregation scheme for a smart grid, while preventing individual user's privacy from leaking to other entities. Our scheme consists of four phases, i.e., *initialization phase, reporting phase, reading phase,* and *dynamic user management*. In the initialization phase, the CC will generate the system parameters and publish them to all users. Then, each user negotiates the secret shared key with a randomly chosen group of users across several BANs. After that, each user chooses another group of users who are in its BAN, and shares a randomly chosen static secret. Once the initialization phase is finished, the reporting phase and reading phase are carried out periodically. In the reporting phase, each user reports encrypted data to the CC. The CC aggregates the reported data and reads the total consumption data in the reading phase. Users can join and leave freely in the smart grid, so a mechanism of dynamic user management is designed to handle the problems of user enrollment, revocation, and malfunction. In our scheme, time is divided into slots with the fixed period. The reporting phase and the reading phase are executed at the beginning of each time slot. The following presents the details of each phase, and the notations used in our scheme are summarized in Table I.

B. Initialization Phase

Generally, we denote the user set for aggregation as $\mathbb{U} = \{U_1, U_2, \ldots, U_N\}$, which can be the users in any user areas. In addition, two different kinds user sets are used in our scheme: 1) $\mathbb{U}^i_{\text{neg}}$ refers U_i chooses a group of users across multiple BANs and 2) $\mathbb{U}^i_{\text{sha}}$ refers U_i chooses a group of users who are in the same BAN with U_i. We use $m_{i,j}$ $(1 \leq i \leq N)$ to present user U_i's metering power data at time slot j. Especially, the initialization phase consists of the following steps.

1) *Public Parameters Generation:* In this step, the CC will generate and publish the public parameters for the system initialization. First, the CC chooses a number $M > N \cdot \max_{i,j}\{m_{i,j}\}$ as the modulus number of homomorphic encryption, which could be an empirical value according to the historical data. Then, the CC generates a superincreasing sequence $b_1, \ldots, b_K$ such that
$$b_1 = 1,\ b_j > M \cdot \sum_{i=1}^{j-1} b_i, \quad j = 2, 3, \ldots, K \qquad (11)$$
where K is the max number of security keys that a user can share with other users. Additionally, the CC chooses a large prime number p which satisfies $p > M \cdot \sum_{i=1}^{K} b_i$, and the threshold value t of the $(t,\ n)$ threshold secret sharing scheme. Then two secure one-way hash functions $H_1(\cdot) : \{0,1\}^* \rightarrow \mathbb{Z}_M$ and $H_2(\cdot) : \{0,1\}^* \rightarrow \mathbb{Z}_p$ are chosen. Finally, the CC publishes the system parameters as
$$\{M, p, K, t, H_1(\cdot), H_2(\cdot), \{b_1, b_2, \ldots, b_K\}\}. \qquad (12)$$
2) *Shared Key Negotiation:* In this step, each user (U_i, $i = 1, 2, \ldots, N$) randomly chooses a set of other users in the system, which can be across more than one BANs, denoted as $\mathbb{U}_{\text{neg}}$. Without loss of generality, we assume $\mathbb{U}^i_{\text{neg}} = \{U_1, U_2, \ldots, U_k\}$ and $|\mathbb{U}^i_{\text{neg}}| = k$. Then U_i generates k shared keys as $\{k_{i1}, k_{i2}, \ldots, k_{ik}\}$, in which each shared key is less than M, and securely distributes them to the corresponding users. Meanwhile, user U_i also receives other users' shared keys (assume the number of

them is k'), i.e., $\{k_{1i}, k_{2i}, \ldots, k_{k'i}\}$. We denote the final shared keys for U_i as the shared key set $\mathbb{K}_{i,0}$

$$\mathbb{K}_{i,0} = \{k_{i1}, k_{i2}, \ldots, k_{ik}; k_{1i}, k_{2i}, \ldots, k_{k'i}\}. \quad (13)$$

It should be noted that the sum of k and k' is limited to no more than K, that is, $|\mathbb{K}_{i,0}| = k + k' \leq K$. This constraint can be satisfied by setting a sufficiently large K, or allowing users to refuse to accept the secret share key from other users.

3) *Static Secret Sharing:* In this step, first, U_i randomly chooses $t-1$ elements $a_{1,i}, a_{2,i}, \ldots, a_{t-1,i} \in \mathbb{Z}_p$, where t is the threshold value of the (t, n) threshold secret sharing scheme. U_i chooses a set of other users in the same BAN, denoted as $\mathbb{U}^i_{\text{sha}}$. Without loss of generality, let $\mathbb{U}^i_{\text{sha}} = \{U_1, U_2, \ldots, U_n\}$, $|\mathbb{U}^i_{\text{sha}}| = n$ where $t \leq n \leq N$. Then, it chooses a random number $ss_i \in \mathbb{Z}_p$ as the static secret and utilizes the (t, n) threshold secret sharing scheme to share the static secret ss_i with users in $\mathbb{U}^i_{\text{sha}}$. In detail, U_i chooses n distinct nonzero elements x_j $(1 \leq j \leq n)$ from $\mathbb{Z}_p$, and computes the shares $(x_j, f(x_j))$ for $U_j \in \mathbb{U}^i_{\text{sha}}$, where $f(x_j) = \sum_{l=0}^{t-1} a_{l,i} x_j^l$ and $a_{0,i} = ss_i$. Finally, U_i securely distributes the shares to corresponding users in $\mathbb{U}^i_{\text{sha}}$.

In summary, the CC generates required system parameters and publishes them to users. Meanwhile, secret share keys and static secrets are generated and shared among the users. Although the initialization phase may introduce some additional communication overhead, normally, this phase only needs to be carried out once.

C. Reporting Phase

In this phase, users encrypt their metering power data, and report to the CC. This phase is presented in the following steps.

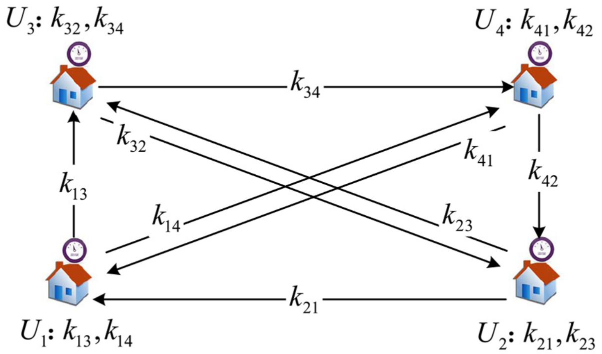
1) *Key Update:* Before reporting data to the CC at the time slot j, $U_i \in \mathbb{U}$ first updates its shared key set to $\mathbb{K}_{ij}$ by computing $\mathbb{K}_{i,j} = H_1(\mathbb{K}_{i,j-1})$, that is, $k_{(\cdot)} = H_1(k_{(\cdot)})$ for each $k_{(\cdot)}$ in $\mathbb{K}_{i,j-1}$, where $\mathbb{K}_{i0}$ is the initialized key set for U_i. Then U_i computes its corresponding dynamic secret S_{ij} using the public superincreasing sequence $\{b_1, b_2, .., b_K\}$ as

$$S_{ij} = \sum_{j=1}^{k} b_i k_{ij} + \sum_{j=1}^{k'} b_{k+j}(-k_{ji}). \quad (14)$$

It should be noted that an additional bit should be appended to denote the negative number $-k_{ji}$, since it cannot be directly encoded with the superincreasing sequence. Then, U_i randomly chooses a number $z_{ij} \in \mathbb{Z}_p$ and computes the parameter pair $(z_{ij}, T_{z_{ij}})$, where $T_{z_{ij}} = S_{ij} - H_2(ss_i \oplus z_{ij})$, and $\oplus$ is bit-wise XOR.

2) *Data Encryption:* Let m_{ij} be U_i's power consumption data at time slot j, where $1 \leq i \leq N$. Before reporting m_{ij}, U_i needs to encrypt m_{ij} by using the homomorphic encryption scheme as (1), that is

$$c_{ij} = Enc_{sk_{ij}}(m_{ij}) = m_{ij} + sk_{ij} \mod M \quad (15)$$



U_1 computes secret key: $sk_1 = k_{13} + k_{14} - k_{21} - k_{41}$

U_2 computes secret key: $sk_2 = k_{21} + k_{23} - k_{32} - k_{42}$

U_3 computes secret key: $sk_3 = k_{32} + k_{34} - k_{13} - k_{23}$

U_4 computes secret key: $sk_4 = k_{41} + k_{42} - k_{14} - k_{34}$

therefore

$$sk_1 + sk_2 + sk_3 + sk_4 = 0$$

Fig. 2. Example for illustrating the correctness of (17).

where

$$sk_{ij} = \sum_{l=1}^{k} k_{il} - \sum_{l=1}^{k'} k_{li} \mod M. \quad (16)$$

3) *Data Reporting:* After the above two steps, U_i sends the parameter pair $(z_{ij}, T_{z_{ij}})$ and the encrypted data c_{ij} to its gateway, which caches $(z_{ij}, T_{z_{ij}})$ for a slot time period, and only reports the c_{ij} to the CC to end this phase.

D. Reading Phase

After receiving the metering data from the users, the CC can obtain the total power consumption during time slot j by aggregating all of the reported data as shown in

$$m_j = \sum_{i=1}^{N} c_{ij} = \sum_{i=1}^{N} m_{ij} + \sum_{i=1}^{N} sk_{ij} \mod M. \quad (17)$$

For clarity, we draw Fig. 2 to give the illustration for the correctness of $\sum_{i=1}^{N} sk_{ij} \mod M = 0$ in (17). It is a system where $N = 4$, $k_i = 2$, and $k'_i = 2$. $\mathbb{U}^1_{\text{neg}} = \{U_3, U_4\}$, $\mathbb{U}^2_{\text{neg}} = \{U_1, U_3\}$, $\mathbb{U}^3_{\text{neg}} = \{U_2, U_4\}$, and $\mathbb{U}^4_{\text{neg}} = \{U_1, U_2\}$. Thus, in time slot j, $\mathbb{K}_{1,j} = \{k_{13}, k_{14}; k_{21}, k_{41}\}$, $\mathbb{K}_{2,j} = \{k_{21}, k_{23}; k_{32}, k_{42}\}$, $\mathbb{K}_{3,j} = \{k_{32}, k_{34}; k_{13}, k_{23}\}$, and $\mathbb{K}_{4,j} = \{k_{41}, k_{42}; k_{14}, k_{34}\}$. In the data reporting phase, users compute their secret keys, respectively, as shown in Fig. 2. Therefore, $\sum_{i=1}^{4} sk_i = 0$. A more general case is also correct as shown in the example. Be informed that the case of a wrong aggregated result due to the wrong encryption key sk_{ij} by U_i will not be discussed in this article. In fact, it can be addressed by using the homomorphic hash function [36] as described in [31].

In each time slot, the reporting phase and reading phase will be executed repeatedly. Based on the aggregated results, CC can learn the power load in real time, which can help improve electricity utilization, develop reasonable dynamic price, and prevent blackouts.

E. Dynamic User Management

From the reading phase, it can be seen that whether the CC can successfully decrypt the aggregated ciphertexts or not depends on whether all users work correctly. That is, the sum of sk_{ij} must be equal to 0, $\sum_{i=1}^{N} sk_{ij} = 0$. But in practice, the users in the smart grid may be malfunctioning because of the limited lifetime of smart meters or natural disaster. In addition, user revocation and new user enrollment can also lead to the change for other users' security key sk_{ij}, which also leads to $\sum_{i=1}^{N} sk_{ij} \neq 0$. To overcome this problem, on the one hand, users of the entire system are divided into multiple aggregation groups according to their user areas. Our proposed scheme can be executed in each aggregation group independently. Therefore, the value of N in our scheme is not so large, and the frequency of user dynamic changes which leads to the possibility that $\sum_{i=1}^{N} sk_{ij} \neq 0$ becomes lower. On the other hand, a dynamic user management mechanism is designed in our scheme. The detailed process is demonstrated as follows.

1) *User Enrollment:* Assume a new user (denoted as U_u) is added to a BAN at time slot j, which is similar to the steps of shared key negotiation and static secret sharing in the initialization phase. First, U_u chooses a user set $\mathbb{U}_{\text{neg}}^{u} = \{U_1, U_2, \ldots, U_k\}$ and negotiates the shared key set $\mathbb{K}_{uj}$. Meanwhile, k' users also choose U_u to share secret keys. Let $\mathbb{K}_{uj} = \{k_{u1}, k_{u2}, \ldots, k_{uk}; k_{1u}, k_{2u}, \ldots, k_{k'u}\}$. Then, U_u chooses another set of other users in the same BAN, assuming $\mathbb{U}_{\text{sha}}^{u} = \{U_1, U_2, \ldots, U_n\}$, generates a static secret $ss_u \in \mathbb{Z}_p$, and shares the static secret ss_u to users in the chosen $\mathbb{U}_{\text{sha}}^{u}$ by Shamir's $(t,\ n)$ threshold secret sharing scheme. Since user U_i $(1 \leq i \leq n)$ in $\mathbb{U}_{\text{neg}}^{u}$ has received additional shared key from the new user U_u, U_i updates his secret shared key set to $\mathbb{K}_{ij} = \{k_{i1}, k_{i2}, \ldots, k_{ik}; k_{1i}, k_{2i}, \ldots, k_{k'i}, k_{ui}\}$, where k_{ui} is the shared key that the new user U_u generates and shares with U_i. The changes of the secret shared key set in this step will take effect in Key Update step of the *reporting phase*, making $\sum_{i\in\mathbb{U}} sk_{ij} = 0$.
2) *User Revocation:* When a user U_r is revoked from the system at time slot j, it cannot send its secret key sk_{ij} to the CC any more. Therefore, $\sum_{i=1,i\neq r}^{N} sk_{ij} \neq 0$ such that the CC cannot access the correct aggregated power data. To overcome this problem, the CC will broadcast a revocation message to U_r's BAN and collect more than t shares of U_r, assume they are $(x_1, f(x_1))$, $(x_2, f(x_2))$, and $(x_t, f(x_t))$. Then, it reconstructs its corresponding static secret ss_r as
$$ss_r = f(0) = \sum_{i=0}^{t} f(x_r) \prod_{j=0,j\neq i}^{t} \frac{-x_j}{x_i - x_j} \mod p. \quad (18)$$
Meanwhile, the CC collects U_r's parameter pair $(z_{rj}, T_{z_{rj}})$ from the BAN gateway. Based on the static secret ss_r and $(z_{rj}, T_{z_{rj}})$, the CC can further compute secret $S_{rj} = T_{z_{rj}} + H_2(ss_r \oplus z_{rj}) \mod p$, and decode it to obtain U_r's secret key set $\mathbb{K}_{rj}$ as (10). Therefore, based on U_r's secret key set $\mathbb{K}_{rj}$, the CC can help revoke user U_r by computing its secret key sk_{rj}, so that the equation $\sum_{i=1}^{N} sk_{ij} = 0$ still holds when U_r is revoked. It should be noted that the above processes are very efficient for users, since all computations are performed on the CC, only several corresponding users need to send their shares to the CC.
3) *User Malfunction:* When a user is malfunctioning, it cannot report its metering data for some periods, but can be recovered to normal after repaired by the CC. In fact, this process can be seen as consisting of a user revocation and a user enrollment. Therefore, when a user U_m is malfunctioning, the CC collects and computes U_m's secret key sk_{mj} as the process of *user revocation*. After U_m has been fixed, U_m acts as a new enrollment user to execute the *user enrollment* process.

VI. Security Analysis

In this section, we discuss the security issues of the proposed scheme via analysis. Our analysis mainly focuses on confidentiality, privacy preservation, forward secrecy, and N-source anonymity.

A. Confidentiality for Metering Data

In the proposed scheme, each user U_i's electricity usage data in time slot t is encrypted as $c_{ij} = m_{ij} + sk_{ij} \mod M$ by smart meters before being submitted to its BAN gateway. We prove its confidentiality as the following lemma.

Lemma 1: The encryption scheme is semantically secure under the assumption that the secure one-way hash function $H_1(\cdot)$ is computational indistinguishability.

Proof: First, we present the definition of the indistinguishability for $H_1(\cdot)$ and $(H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_M)$. Given a fixed input argument r, the output $y = H_1(r)$ is computationally indistinguishable from a randomly picked number $y \leftarrow \mathbb{Z}_M$. Formally, for any polynomial time adversary $\mathcal{A}$
$$|Pr[y = H_1(r) : \mathcal{A}(x,y) = 1] \quad (19)$$
$$-Pr[y \leftarrow \mathbb{Z}_M : \mathcal{A}(x,y) = 1]| < \epsilon(\lambda) \quad (20)$$
where $\epsilon(\lambda)$ is a negligible function in λ.

Then, we use the IND-CPA game to prove our lemma. Proving the lemma is the same as proving if our encryption scheme is not secure. Then, the assumption that $H_1(\cdot)$ is indistinguishable does not hold. We assume a polynomial time adversary $\mathcal{A}$ against our scheme, then we can construct an adversary to distinguish $H_1(\cdot)$ as follows.

1) $\mathcal{A}$ chooses two messages m_0 and m_1.
2) The challenger chooses $b \in \{0,1\}$, and outputs $c = m_b + sk \mod M$. Meanwhile, we choose a random number r, compute $x = H_1(r)$, and want to find out whether x and c are distinguishable. Then, we compute $c' = c + x$ and send c' to $\mathcal{A}$.
3) $\mathcal{A}$ decides if c' is an encryption of m_0 or m_1, and answers $b' \in \{0,1\}$. If $b' = b$, we say $H_1(\cdot)$ is distinguishable, otherwise is indistinguishable.

Therefore, there are two cases described as follows.

1) *x Is Distinguishable:* c' is an encryption of m_0 or m_1. Thus, $\mathcal{A}$ has a nonnegligible advantage. It means that it has a probability nonnegligible more than 1/2 to choose

the right one. When $\mathcal{A}$ wins, we win as well. In other words, we also have a probability nonnegligibly higher than 1/2 to distinguish x.

2) *x Is Indistinguishable:* c' is an encryption of random message. Thus, $\mathcal{A}$ chooses with zero advantage, which means it has only probability 1/2 to choose the right one. So when $\mathcal{A}$ wins, we lose and this happens with probability 1/2.

In summary, we more often choose right, when x is distinguishable, than wrong, when x is indistinguishable. So we can distinguish a secure hash function with probability nonnegligibly more than 1/2. In other words, our proposed encryption is semantically secure if the secure one-way hash function $H_1(\cdot)$ is computational indistinguishability. Thus, the confidentiality of metering data is protected. ■

B. Privacy Preservation for Users

In order to prevent the privacy of users from disclosing, other parties (the CC or the gateway) are not allowed to recover each individual customer's usage data. Here, we prove this goal by the following lemmas.

Lemma 2: Suppose that the number of system users satisfies $|\mathbb{U}| \geq 2$ and the CC cannot compromise each individual user's security key sk_{ij} $(U_i \in \mathbb{U})$. The CC cannot violate the individual user's privacy.

Proof: First, from Lemma 1, we can know that the CC is unable to access each individual user's metering data without knowing its security key. Although, the CC can read the aggregated data $\sum_{U_i \in \mathbb{U}} m_{ij}$ from the ciphertext, as shown in (17), it is still impossible for the CC to infer individual user's metering data from the aggregated result $\sum_{U_i \in \mathbb{U}} m_{ij}$ since $|\mathbb{U}| \geq 2$. Therefore, without any individual user's security key sk_{ij}, it is impossible for the CC to access individual user's electricity usage data. ■

Lemma 3: Suppose the CC colludes with the user set $\mathbb{U}^c$, where $|\mathbb{U}^c| < \min\{k + k', t\} < |\mathbb{U}|$, the CC is unable to infer other noncolluding individual user's metering data.

Proof: By colluding with user set $\mathbb{U}^c$, the CC can obtain the aggregated result $\sum_{U_i \in \mathbb{U}^c} m_{ij}$ and $\sum_{U_i \in \mathbb{U}} m_{ij}$. Since $|\mathbb{U}| - |\mathbb{U}^c| \geq 2$, the CC cannot infer other noncolluding individual user's metering data as proving in Lemma 2. On the other hand, the CC may try to recover the security key sk_{ij} from the following two ways.

1) The CC tries to learn each shared key in $\mathbb{K}_{ij}$ and computes $sk_{ij} = \sum_{l=1}^{k} k_{il} - \sum_{l=1}^{k'} k_{li}$. However, since the collusion user set $\mathbb{U}^c < k + k'$, the CC is unable to compute sk_{ij} by this way.
2) In our scheme, a user's shared key set $\mathbb{K}_{ij}$ is also shared by the $(t,\ n)$ threshold secret sharing scheme. The CC may collect shares by colluding with users, and computes sk_{ij} by (16). However, under the collusion with less than t participants [32] and $|\mathbb{U}^c| < t$, the search space of a_0, which is each user's static secret, will not be reduced. Therefore, it is still impossible for the CC to recover sk_{ij} through this way.

In summary, when the colluding user set $|\mathbb{U}^c| < \min\{k + k', t\} < |\mathbb{U}|$, CC is unable to infer other noncolluding individual user's metering data. ■

TABLE II
SECURITY COMPARISON WITH RELATED SCHEMES

Scheme	Data Confidentiality	No Trusted Authority	Dynamic Users	Forward Secrecy	N-source Anonymity
[15]	Yes	Yes	No	No	Yes
[14]	Yes	No	No	Yes	Yes
[31]	No	Yes	Yes	Yes	No
Our Scheme	Yes	Yes	Yes	Yes	Yes

C. Forward Secrecy

Lemma 3: Our scheme can achieve the forward secrecy of a user's security key when the CC does not collude with any gateways.

Proof: At each time slot, each user executes the key update step. At this step, each shared key in $\mathbb{K}_{ij}$ will update by using a secure one-way hash function $H_1(\cdot)$, that is, $\mathbb{K}_{i,j} = H_1(\mathbb{K}_{i,j-1})$ for each $k_{(\cdot)}$ in $\mathbb{K}_{i,j-1}$. Meanwhile, the parameter pair $(z_{ij}, T_{z_{ij}})$ which is used to recover user's security key also updates periodically. Therefore, even a user's security key is compromised, or recovered by the CC when it is revoked, it is infeasible to access the user's previous ciphertexts. Thus, the forward secrecy is achieved in our scheme.

It should be noted that the forward security will not be ensured if the gateways collude with the CC. Once the gateways collude with the CC, the CC can get $(z_{rj}, T_{z_{rj}})$ of any time slot. When user U_r is revoked from the system or malfunctions at time slot j, the CC can reconstruct its corresponding static secret ss_r. Thus, U_r's data before time slot j will be disclosed to the CC. ■

D. N-Source Anonymity

Lemma 4: Let N be the number of users that do not collude with the CC in the smart grid system, our proposed scheme can achieve N-source anonymity.

Proof: During the step of shared key negotiation in the initialization phase, a user U_i is allowed to negotiate shared keys with other users in the whole system, not only in its BAN. Meanwhile, each user computes its security key based on their negotiated shared keys as $sk_{ij} = \sum_{l=1}^{k} k_{il} - \sum_{l=1}^{k'} k_{li}$. Therefore, only the sum of the security keys for all users in the system is equal to 0, that is, $\sum_{i=0}^{N} sk_{ij} = 0$. In other words, the CC is only allowed to obtain N users' aggregated result. Thus, our scheme achieves the security property of N-source anonymity. ■

E. Security Feature Comparisons

We compare the proposed scheme with several other representative masking-based privacy-preserving data aggregation scheme for a smart grid in terms of the aspects of data confidentiality, trusted authority, dynamic user management, forward secrecy, and N-source anonymity. As shown in Table II, these existing works have not yet proposed an ideal data aggregation scheme with all of the desired properties.

VII. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed scheme in terms of computation, communication, and storage overhead.

A. Computation Overhead

In the real-time data aggregation paradigm of the smart grid, data are frequently reported to the CC. Therefore, the performance of the reporting phase and reading phase is our main concern. To specifically evaluate our scheme's performance, we compare our scheme with three representative related works, *viz.*, CommEffi [15], ErrorRes [31], and Paillier-based scheme. The Paillier-based scheme refers to such a type of scheme, such as [7] and [13], which used the Paillier cryptosystem to protect users' privacy in data aggregation. Then we conduct the experiments in a laptop with the Intel Core i5-3230M CPU @2.60GHz and 4GB RAM, based on the GMP library [37] and OpenSSL library [38]. The experimental results are shown in Fig. 3.

1) Computation Overhead in the Reporting Phase: In the reporting phase, each user encrypts its data and sends the encrypted data to the CC. Therefore, the running time in this phase is the cost in the user side. Fig. 3(a) shows the comparison of computation overhead in this phase. To be noted, we use the form of a subfigure within a figure to clearly show the local variation. From the figure, it can be seen that the running time of the Paillier-based scheme is obviously larger than other masking-based schemes (CommEffi [15], ErrorRes [31], and our scheme), since it uses a public key-based homomorphic encryption. When compared to other masking-based schemes, the computation overhead of our scheme is much more efficient than CommEffi [15], and a bit more efficient than ErrorRes [31]. It should be noted that the efficiency of this phase is very important for the actual deployment of the protocol to the smart meters in limited computing power.

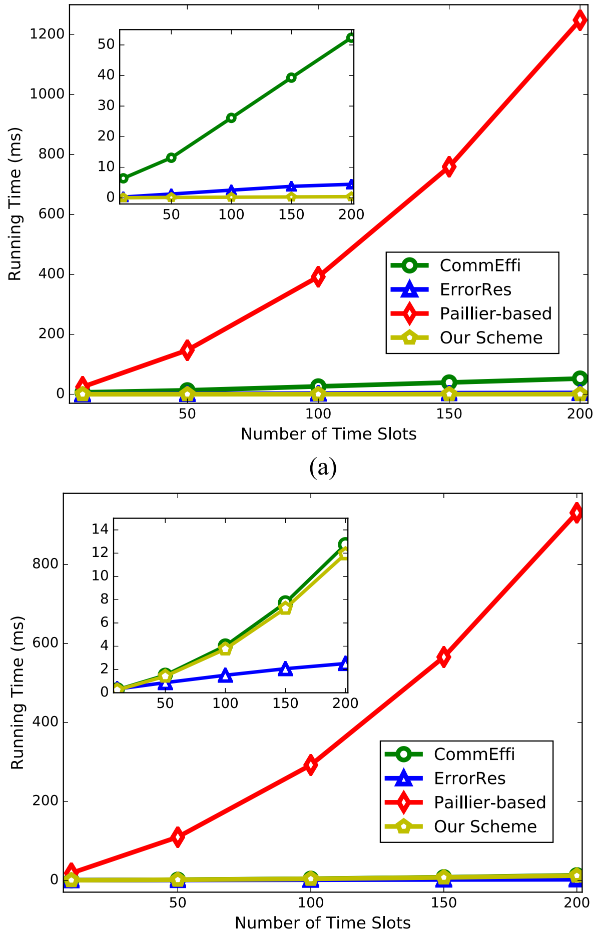
2) Computation Overhead in the Reading Phase: In the reading phase, the overhead of computation occurs in the CC side for decrypting the reporting data. As shown in Fig. 3(b), the computation overhead of our scheme can be seen as negligible when compared to the Paillier-based scheme, while the running time of our scheme is closed to CommEffi [15], and a little more than ErrorRes [31].

B. Communication Overhead

In this part, we evaluate the communication overhead with representative related works: CommEffi [15], ErrorRes [31], and Paillier-based scheme (such as [7] and [13]). Specifically, the size of ciphertexts encrypted by Paillier is set as 2048 bits, ciphertexts encrypted by the masking approach is set as 32 bits. We set the number of time slots as a fixed number 100, and $|p| = 512$ bits. The results of communication overhead are presented in Table III. From the table, it can be seen that the communication overhead of our scheme is more efficient than the Paillier-based scheme, and a little larger than CommEffi and ErrorRes. However, it is acceptable since it only costs $1.25 * 10^{-4}$ MB for each user at a time slot.

C. Storage Overhead

In our scheme, users need to store system parameters (i.e., $\{a_{1,i}, a_{2,i}, \ldots, a_{t-1,i}\}$ and $\{b_1, b_2, \ldots, b_K\}$) and shared security key set $\mathbb{K}_{ij}$. It would bring some storage overhead to our scheme. To evaluate the storage overhead, we draw Table IV and compare the overhead with related works in terms of the number of the system users, where we set $t = K = 20$, and the length of user identity is 16 bits. From the table, we can see that the storage overhead of our scheme is fixed in terms of the number of system users, the same as the Paillier-based scheme, and it is more efficient than CommEffi [15]



(a)

(b)

Fig. 3. Comparison of computation overhead with related works. (a) Comparison for the reporting phase. (b) Comparison for the reading phase.

TABLE III
COMPARISON OF COMMUNICATION OVERHEAD

Overhead(MB)	CommEffi	ErrorRes	Paillier-based	Our Scheme
100 Users	0.04	0.23	2.44	1.26
500 Users	0.19	1.14	12.21	6.29
3000 Users	1.14	6.87	73.24	37.77
10000 Users	3.81	22.89	244.14	125.88

TABLE IV
COMPARISON OF STORAGE OVERHEAD

Overhead(KB)	CommEffi	ErrorRes	Paillier-based	Our Scheme
100 Users	0.39	0.20	0.25	1.41
500 Users	1.95	0.98	0.25	1.41
3000 Users	11.72	5.87	0.25	1.41
10000 Users	39.06	19.54	0.25	1.41

and ErrorRes [31] when the number of system users increases.

VIII. Conclusion

In this article, we proposed an efficient, privacy friendly, and robust data aggregation scheme for the smart grid. Our design does not need a trusted authority and it supports flexible dynamic user management. Analysis of security and performance shows that the design goals can be well satisfied. In our future work, we will try to give a more lightweight scheme to further reduce the overhead of user revocation and user malfunction.

Acknowledgment

The authors would like to thank all the anonymous reviewers for their valuable suggestions that have led to the present improved version of this article.

References

[1] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3557–3564, Oct. 2010.

[2] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid—The new and improved power grid: A survey," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 944–980, 4th Quart., 2012.

[3] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Security Privacy*, vol. 8, no. 1, pp. 81–85, Jan./Feb. 2010.

[4] S. Li, K. Xue, D. S. L. Wei, H. Yue, N. Yu, and P. Hong, "SecGrid: A secure and efficient SGX-enabled smart grid system with rich functionalities," *IEEE Trans. Inf. Forensics Security*, to be published, doi: 10.1109/TIFS.2019.2938875.

[5] Y. Liu, W. Guo, C.-I. Fan, L. Chang, and C. Cheng, "A practical privacy-preserving data aggregation (3PDA) scheme for smart grid," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1767–1774, Mar. 2019.

[6] J. Song, Y. Liu, J. Shao, and C. Tang, "A dynamic membership data aggregation (DMDA) protocol for smart grid," *IEEE Syst. J.*, to be published.

[7] 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 Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.

[8] A. Abdallah and X. S. Shen, "A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 396–405, Jan. 2018.

[9] L. Zhu *et al.*, "Privacy-preserving authentication and data aggregation for fog-based smart grid," *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 80–85, Jun. 2019.

[10] Q. Yang, J. Hong, K. Xue, W. Chen, X. Zhang, and H. Yue, "A privacy-preserving and real-time traceable power request scheme for smart grid," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, 2017, pp. 1–6.

[11] S. Li, X. Zhang, K. Xue, L. Zhou, and H. Yue, "Privacy-preserving prepayment based power request and trading in smart grid," *China Commun.*, vol. 15, no. 4, pp. 14–27, 2018.

[12] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: Privacy-preserving multisubset data aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 462–471, Feb. 2018.

[13] K. Xue *et al.*, "PPSO: A privacy-preserving service outsourcing scheme for real-time pricing demand response in smart grid," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2486–2496, Apr. 2019.

[14] 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 Things J.*, vol. 5, no. 4, pp. 3126–3135, Aug. 2018.

[15] X. Gong, Q.-S. Hua, L. Qian, D. Yu, and H. Jin, "Communication-efficient and privacy-preserving data aggregation without trusted authority," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, 2018, pp. 1250–1258.

[16] P. Gope and B. Sikdar, "Lightweight and privacy-friendly spatial data aggregation for secure power supply and demand management in smart grids," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 6, pp. 1554–1566, Jun. 2019.

[17] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: Privacy preserving fog-enabled aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3733–3744, Aug. 2018.

[18] F. M. Tabrizi and K. Pattabiraman, "A model for security analysis of smart meters," in *Proc. IEEE/IFIP 42nd Int. Conf. Depend. Syst. Netw. Workshops (DSN-W)*, Boston, MA, USA, 2012, pp. 1–6.

[19] N. Saxena and B. J. Choi, "State of the art authentication, access control, and secure integration in smart grid," *Energies*, vol. 8, no. 10, pp. 11883–11915, 2015.

[20] T. W. Chim, S.-M. Yiu, L. C. K. Hui, and V. O. K. Li, "PASS: Privacy-preserving authentication scheme for smart grid network," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Brussels, Belgium, 2011, pp. 196–201.

[21] T. W. Chim, S.-M. Yiu, V. O. K. Li, L. C. K. Hui, and J. Zhong, "PRGA: Privacy-preserving recording & gateway-assisted authentication of power usage information for smart grid," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 1, pp. 85–97, Jan./Feb. 2015.

[22] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Tech. (EUROCRYPT)*, 1999, pp. 223–238.

[23] 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 Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2053–2064, Aug. 2014.

[24] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proc. Theory Cryptography Conf.*, 2005, pp. 325–341.

[25] H. Bao and R. Lu, "A new differentially private data aggregation with fault tolerance for smart grid communications," *IEEE Internet Things J.*, vol. 2, no. 3, pp. 248–258, Jun. 2015.

[26] L. Chen, R. Lu, Z. Cao, K. AlHarbi, and X. Lin, "MuDA: Multifunctional data aggregation in privacy-preserving smart grid communications," *Peer-to-Peer Netw. Appl.*, vol. 8, no. 5, pp. 777–792, 2015.

[27] 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 Trans. Smart Grid*, vol. 8, no. 5, pp. 2483–2493, Sep. 2017.

[28] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Trans. Sensor Netw. (TOSN)*, vol. 5, no. 3, p. 20, 2009.

[29] H. Mohammed, S. Tonyali, K. Rabieh, M. Mahmoud, and K. Akkaya, "Efficient privacy-preserving data collection scheme for smart grid AMI networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1–6.

[30] S. Tonyali, O. Cakmak, K. Akkaya, M. M. E. A. Mahmoud, and I. Guvenc, "Secure data obfuscation scheme to enable privacy-preserving state estimation in smart grid AMI networks," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 709–719, Oct. 2016.

[31] F. Knirsch, G. Eibl, and D. Engel, "Error-resilient masking approaches for privacy preserving data aggregation," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3351–3361, Jul. 2018.

[32] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[33] R. C. Merkle, "Secure communications over insecure channels," *Commun. ACM*, vol. 21, no. 4, pp. 294–299, 1978.

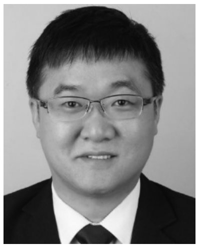
[34] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient Merkle-tree-based authentication scheme for smart grid," *IEEE Syst. J.*, vol. 8, no. 2, pp. 655–663, Jun. 2014.

[35] L. Sweeney, "*k*-anonymity: A model for protecting privacy," *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[36] M. N. Krohn, M. J. Freedman, and D. Mazieres, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *Proc. IEEE Symp. Security Privacy (S&P)*, Berkeley, CA, USA, 2004, pp. 226–240.

[37] *The GNU MP Bignum Library*. Accessed: Jan. 9, 2020. [Online]. Available: https://gmplib.org/

[38] *OpenSSL: Cryptography and SSL/TLS Toolkit*. Accessed: Jan. 9, 2020. [Online]. Available: https://www.openssl.org/



Kaiping Xue (Senior Member, IEEE) received the B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), Hefei, China, in 2003, and the Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC in 2007.

From May 2012 to May 2013, he was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. He is currently an Associate Professor with the School of Cyber security, Department of EEIS, USTC. His research interests include next-generation Internet, distributed networks, and network security.

Dr. Xue serves on the Editorial Board of several journals, including the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, *Ad Hoc Networks*, IEEE ACCESS, and *China Communications*. He has served as a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and a Lead Guest Editor for *IEEE Communications Magazine*. He is an IET Fellow.

Bin Zhu received the B.S. degree in information security from the School of Cyber Security, University of Science and Technology of China, Hefei, China, in 2019, where he is currently pursuing the graduation degree in information security with the School of Cyber Security.

His research interests include privacy-preserving computing and applied cryptography.

Qingyou Yang received the B.S. degree in information security from the School of the Gifted Young, University of Science and Technology of China (USTC), Hefei, China, in 2016, and the M.S. degree from the Department of Electronic Engineering and Information Science, USTC, in 2019.

He is currently an Engineer with Baidu, Shenzhen, China. His research interests include network security and cryptography.

David S. L. Wei (Senior Member, IEEE) received the Ph.D. degree in computer and information science from the University of Pennsylvania, Philadelphia, PA, USA, in 1991.

From May 1993 to August 1997, he was an Associate Professor and then a Professor with the Faculty of Computer Science and Engineering, University of Aizu, Aizuwakamatsu, Japan. He has authored and coauthored more than 100 technical papers in various archival journals and conference proceedings. He is currently a Professor with the Computer and Information Science Department, Fordham University, New York, NY, USA. His research interests include cloud computing, big data, IoT, and cognitive radio networks.

Prof. Wei was a Guest Editor or a Lead Guest Editor for several special issues in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, and the IEEE TRANSACTIONS ON BIG DATA. He also served as an Associate Editor for the IEEE TRANSACTIONS ON CLOUD COMPUTING from 2014 to 2018, and *Journal of Circuits, Systems and Computers* from 2013 to 2018.

Mohsen Guizani (Fellow, IEEE) received the B.S. (Distinction) and M.S. degrees in electrical engineering, and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively.

He served in different academic and administrative positions with the University of Idaho, Moscow, ID, USA, Western Michigan University, Kalamazoo, MI, USA, University of West Florida, Pensacola, FL, USA, University of Missouri-Kansas City, Kansas City, MO, USA, University of Colorado–Boulder, Boulder, CO, USA, and Syracuse University. He is currently a Professor with the CSE Department, Qatar University, Doha, Qatar. He has authored nine books and more than 500 publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid.

Prof. Guizani received three teaching awards and four research awards, the 2017 IEEE Communications Society WTC Recognition Award and as well as the 2018 Ad Hoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor Networks. He guest edited a number of special issues in IEEE journals and magazines. He also served as a member, the Chair, and the General Chair for a number of international conferences. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He is currently the Editor-in-Chief of *IEEE Network Magazine*, serves on the editorial boards of several international technical journals and is the Founder and the Editor-in-Chief of *Wireless Communications and Mobile Computing Journal* (Wiley). He served as the IEEE Computer Society Distinguished Speaker. He is currently the IEEE ComSoc Distinguished Lecturer. He is a Senior Member of ACM.