Fault-Tolerant Multisubset Aggregation Scheme for Smart Grid

Xiaodi Wang, Yining Liu, and Kim-Kwang Raymond Choo

***Abstract*—As smart cities and nations are fast becoming a reality, so does the underpinning infrastructure, such as smart grids. One particular challenge associated with smart grid implementation is the need to ensure privacy preserving multisubset data aggregation. Existing approaches generally require the collaboration of a trusted third party (TTP), which may not be practical. This also increases the threat exposure, as the attacker can now target the TTP who may be servicing several smart grid operators. Therefore, in this article, a fault-tolerant multisubset data aggregation scheme is proposed. Our scheme aggregates the total electricity consumption value, and obtains the number of users and the total electricity consumption in different numerical intervals, without relying on any TTP. Detailed system analysis shows that our scheme prevents the leakage of single data, as well as guarantees the efficiency when new user joins and existing user leaves. Findings from our evaluation also demonstrate that system robustness is achieved with negligible cost.**

***Index Terms*—Fault tolerance, multisubset aggregation, privacy, smart grid.**

I. Introduction

As OUR society becomes "smarter," more of our infrastructure (including those in the critical infrastructure sectors) will also transition to smart, Internet-connected systems. Examples include smart grids, which are known to offer a broad range of benefits over conventional power grids, for instance, in terms of communication and data analytical capabilities. However, there are also privacy considerations, perhaps except for authoritarian countries. For example, the analysis of individual user consumption data can help to inform real-time charging demands and pricing strategy. However, we need to ensure that such benefits do not come at the cost of infringing user privacy or safety. To strike a balance data utility and data privacy, one can use data aggregation solutions. In detail, smart meters (SMs) with limited resources periodically collect and send processed data to the aggregator (AG), for example, using some secure and efficient encryption technique. The AG then computes the sum of the received data, whose result is generally not "interpretable" by the AG. In other words, only the data center (DC) can decrypt the aggregated ciphertext and obtain the sum of the user's raw data [1]–[3].

The interest in data aggregation is evidenced by the number of schemes presented in the literature [4]–[6]. In [7], for example, a privacy-preserving data aggregation scheme based on differential privacy to resist human-factor-aware attacks was presented. Vahedi *et al.* [8] proposed a secure data aggregation to resist internal attackers. However, in their approach, only the sum in the aggregation area can be obtained. In addition, initialization operation needs to be executed whenever a new user joins, which is not practical in a deployment where the user needs to join or exit frequently. In [9], a simple and efficient data aggregation scheme was proposed, and the authors in [10] presented an efficient data aggregation for mobile edge computing that achieves low latency and fast access for an Internet of Things (IoT) setup. In [11], an efficient privacy-preserving protocol for smart metering systems was presented.

In addition to privacy, there are also other desirable features such as multisubset aggregation. In 2015, for example, Erkin [12] used the Chinese Remainder Theorem and homomorphic encryption to implement data aggregation. Their approach uses statistics of user groups in different regions in their computation, while seeking to preserve user's privacy. There have been several other multidimensional aggregation schemes (e.g., [13]), as well as subset data aggregation schemes designed for fine-grained analysis and decision-making [14], [15]. Supporting users dynamic join and exit is also another essential requirement. The virtual aggregation domain was proposed in 3PDA [16], which allows users to join the data aggregation dynamically. In [17], false data injection attack resistance is achieved using the blinding factor. However, it is vulnerable to collusion attacks from control centers and fog devices, and cannot support users dynamic join and exit. In [18], a privacy-preserving truth discovery scheme was introduced, which relies on two noncolluding cloud platforms in the crowd sensing system. However, it is not designed for situations where users need to join and exit frequently.

Manuscript received July 8, 2020; accepted August 2, 2020. Date of publication August 5, 2020; date of current version March 5, 2021. This work was supported in part by the Natural Science Foundation of China under Grant 61662016, in part by the Key projects of Guangxi Natural Science Foundation under Grant 2018JJD170004, and in part by the Study Abroad Program for Graduate Student of Guilin University of Electronic Technology under Grant GDYX2019029. Paper no. TII-20-3312. *(Corresponding author: Yining Liu.)*

Xiaodi Wang and Yining Liu are with the Guangxi Key Laboratory of Trusted Software, School of Computer and Information Security, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: wangxiaodi2019@gmail.com; ynliu2011@gmail.com).

Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249-0631 USA (e-mail: raymond.choo@fulbrightmail.org).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2020.3014401

Some solutions require the assistance of a trusted third party (TTP) to distribute relevant security parameters. There have been many schemes designed based on the smart metering architecture presented in [19]. For example, a decentralized efficient privacy-preserving and selective aggregation scheme with a multirecipient model was introduced in [20]. More recently, in 2019, Mustafa *et al.* [21] proposed another secure and privacy-preserving protocol in the multirecipient model, based on the combination of both multiparty computation and the linear secret sharing. Following [21] a privacy-preserving data collection and access control scheme named EPDA was proposed by Alsharif *et al.* [22], which not only protects individual user's personal consumer data (PCD) but also hides the customers' distribution of each supplier. Similarly, Lu *et al.* [23] proposed a novel privacy-preserving set aggregation scheme to achieve two-subset aggregation from single aggregated data. However, the approach requires the distribution of blinding factor through an offline TTP. In [24], a human-factor-aware privacy-preserving scheme designed to resist human-factor-aware differential aggregation attacks was proposed. It also involves a TTP. Li *et al.* [25] proposed a privacy-preserving multisubset data aggregation (PPMA), where user power data of different ranges are aggregated. PPMA relies on a TTP in the generation and distribution of relevant security parameters. However, reliance on a fully trusted entity in an IoT environment reduces the robustness of the system and increases the maintenance cost.

Hence, there have been attempts to design schemes without relying on any TTP. Ni *et al.* [26] presented an efficient data aggregation without involving any TTP, and user's privacy is preserved even if all entities in the system are not fully trusted. He *et al.* [27] proposed a privacy-preserving multifunction data aggregation scheme, also without a TTP. The protocol utilizes Shamir's secret sharing to allow SMs to negotiate aggregation parameters. In 2019, for example, Xue *et al.* [28] proposed a privacy-preserving service outsourcing (PPSO) scheme for real-time pricing demand response in the smart grid. PPSO is designed to achieve fault tolerance and facilitate flexible customers' enrollment and revocation. However, it cannot resist attacks from the power utility (i.e., insider attacks). Such attacks can have devastating consequences, and are generally more challenging to detect and prevent [29]. As an insider attack may go undetected much longer than attacks from external sources, the consequences (e.g., legal and financial implications) will also be severe. Therefore, we posit the importance of designing aggregation schemes to be resilient to insider attacks.

In this article, a fault-tolerant multisubset aggregation (FTMA) scheme is proposed. FTMA not only eliminates the reliance on TTP, but also allows the number of users and the total power consumption in each area to be obtained and has the following properties:

1) *Nonreliance on TTP:* This minimizes or eliminates performance bottleneck, as well as the need to trust an external entity.
2) *Privacy:* The aggregation value is obtained and published, while preserving individual user privacy (since individual data is not available to other users/consumers).
3) *Fault tolerance:* The scheme is executed even if some SMs are nonfunctional. Maximum utilization of the collected data is achieved, and the data of the faulty user is omitted.
4) *Dynamic entry and exit:* The complicated initialization is not repeatedly executed, when a new user joins or an existing user leaves.
5) *Insider attack resiliency:* The proposed scheme is secure against the attacks from system insider (including DC).

The rest of this article is organized as follows. In Section II and III, the preliminaries (i.e. Paillier cryptosystem, bilinear pairing, and assumptions), and the system and privacy attacker models are introduced, respectively. Section IV presents our proposed protocol, is presented in Section IV, followed by its security and performance evaluation in Section V. Finally, Section VI concludes this article.

II. Preliminaries

Let G_1, G_2 be a cyclic addition group on elliptic curve and a cyclic multiplication with the same order q, where q is a large prime and P_0 is a generator of G_1. $\hat{e} : G_1 \times G_1 \rightarrow G_2$ satisfies the following properties (see also [17]):

1) *Bilinearity:* $\hat{e}(xP, yQ) = \hat{e}(P,Q)^{xy}$ for $\forall x, y \in \mathbb{Z}_q^*$, $\forall P, Q \in G_1$. ($\mathbb{Z}_q^*$ is a set $\{1, 2, \ldots, q-1\}$; more details can prefer the reference).
2) *Nondegeneracy:* $\hat{e}(P_0, P_0) \neq 1_{G_1}$.
3) *Computability:* for all $P, Q \in G_1$, there is an efficient algorithm to compute $\hat{e}(P,Q)$.

Paillier cryptosystem [25] comprises three algorithms:

1) *Key generation:* Selects two large prime number p and q, and then computes $\lambda = \mathrm{lcm}(p-1, q-1)$ and $N = pq$, the function (lcm means take the least common multiple of two numbers and $|p| = |q| = l$ ($|p|$ and $|q|$ is the length of p and q), assuming l is 512 bits. Define function $L(x) = \frac{x-1}{N}$, choose a generator $g \in \mathbb{Z}_{N^2}^*$, and compute $\mu = (L(g^\lambda \bmod N^2))^{-1} \bmod N$. The public key is (N, g) and the private key is (λ, μ).
2) *Encryption:* Given a message $m \in \mathbb{Z}_N^*$, selects a random number $r \in \mathbb{Z}_{N^2}^*$ and calculates the ciphertext $c = E(m) = g^m \cdot r^N \bmod N^2$.
3) *Decryption:* Given ciphertext $c \in \mathbb{Z}_{N^2}^*$, computes the plaintext as: $m = L(c^\lambda \bmod N^2) \cdot \mu \bmod N^2$.

In addition, two assumptions are used in our article:

Assumption 1: (DBDH: Decisional Billinear Diffie-Hellman assumption [30]). Given $P_0, aP_0, bP_0 \in G_1$ for unknown $a, b \in \mathbb{Z}_q^*$ and $R \in G_2$, deciding whether $\hat{e}(P_0, P_0)^{ab} = R$.

Assuming that the DBDH assumption holds in (G_1, G_2) for any probabilistic polynomial-time (PPT) algorithm $\mathbf{B}$, the following condition is true:

$$\begin{aligned}&\left| \Pr\left[a, b \leftarrow \mathbb{Z}_q^*; 1 \leftarrow \mathrm{B}\left(P_0, aP_0, bP_0, e(P_0, P_0)^{ab}\right)\right] \right.\\ &\left. - \Pr\left[a, b \leftarrow \mathbb{Z}_q^*; R {\leftarrow_R} G_2; 1 \leftarrow \mathrm{B}\left(P_0, aP_0, bP_0, R\right)\right] \right|.\\ &\leq v\left(\gamma\right)\end{aligned}$$

In the above equation, $v(\cdot)$ is a negligible function, i.e., $v(\cdot) < 1/\operatorname{poly}(\gamma)$ for every polynomial function poly $(\cdot)$, and γ is a parameter to determine the computational complexity

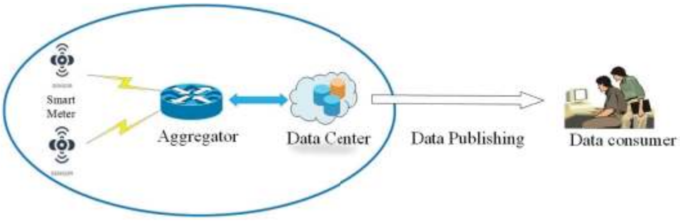

Fig. 1. Design goals.

or security level with sufficient size that reduces the risk of brute-force attack.

Assumption 2: (CDHA: Computational Diffie–Hellman Assumption). Given $P_0, aP_0, bP_0 \in G_1$ where $a, b \in \mathbb{Z}_q^*$, it is computationally infeasible to compute $abP_0 \in G_1$.

III. Models

The system model and privacy attacker model are introduced in Sections III-A and III-B, respectively.

A. System Model

The system includes three entities: DC, AG, and SM.

1) SM: Each user equipped with SM collects the electricity data and sends the encrypted data to AG. In this article, user and SM are used interchangeably.
2) AG: AG receives and aggregates the data from n users, and sends the result to DC. For instance, it could be an edge server or a fog node.
3) DC: DC owns the computing and storage capabilities of generating and distributing security parameters, and decrypting the data from AG. Also, DC publishes or shares the privacy-preserving data to data consumer.

B. Privacy Attacker Model

In traditional attacker models, the attacker is assumed to carry out various of attacks such as denial of service (DoS) attacks, eavesdropping attacks, man-in-the-middle attacks, tampering attacks, impersonation attacks, and replay attacks. However, in this article, we assume that there exists secure communication channel between the authorized entities, and hence we only consider privacy attacker in our model (see Fig. 1).

As discussed earlier, privacy is a crucial consideration in data publishing, and a key threat to privacy is the insider [1], [21]. In our context, AG, DC, or SM are all potential privacy attackers in the sense that they may attempt to deduce useful information from the messages they receive although they are authorized entities in the data aggregation system. We assume that the insider attacker $\mathscr{A}$ can launch multiple attacks, and $\mathscr{A}$ may also monitor the communications among the entities in the system. In addition, $\mathscr{A}$ may attempt to compromise other users, in their attempts to obtain the corresponding security parameters and other useful information about these users. However, there are some basic considerations for the privacy attack assumption:

1) System entities including AG, DC, and SM have a common interest of publishing the privacy-preserving data.

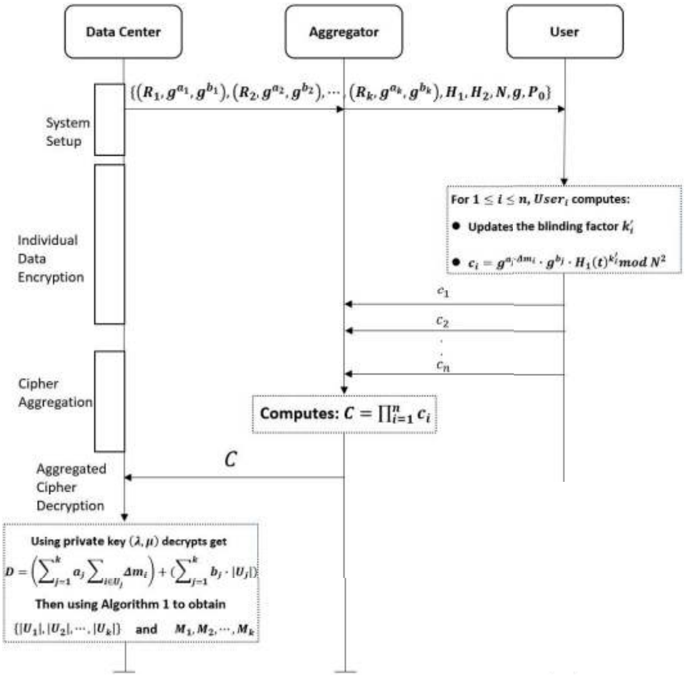
Data Center
Aggregator
User
$\{(R_1, g^{a_1}, g^{b_1}), (R_2, g^{a_2}, g^{b_2}), \cdots, (R_k, g^{a_k}, g^{b_k}), H_1, H_2, N, g, P_0\}$
System Setup
For $1 \le i \le n$, $User_i$ computes:
● Updates the blinding factor k_i'
● $c_i = g^{a_j \cdot \Delta m_i} \cdot g^{b_j} \cdot H_1(t)^{k_i'} mod\ N^2$
Individual Data Encryption
c_1
c_2
c_n
Cipher Aggregation
Computes: $C = \prod_{i=1}^{n} c_i$
C
Aggregated Cipher Decryption
Using private key (λ, μ) decrypts get
$D = \left(\sum_{j=1}^{k} a_j \sum_{i \in U_j} \Delta m_i\right) + (\sum_{j=1}^{k} b_j \cdot |U_j|)$
Then using Algorithm 1 to obtain
$\{|U_1|, |U_2|, \cdots, |U_k|\}$ and $M_1, M_2, \cdots, M_k$

Fig. 2. Proposed faulttolerant multisubset data aggregation (FTMA) scheme.

2) AG, DC, and SMs are all authorized entities, and any fabricated/injected/modified message can be easily identified and punished [31].

IV. Our Proposed FTMA Scheme

In this section, our FTMA scheme without TTP is presented, which consists of five phases: system setup, individual data encryption, cipher aggregation, aggregated cipher encryption, and fault tolerance mechanism (see also Fig. 2).

A. System Setup

The electricity consumption data is classified into k continuous subsets, $[R_1, R_2), [R_2, R_3), \ldots, [R_k, E]$, where $R_1 = 0$, $E = \max$, and $m_i \in [R_i, R_{i+1})$ indicates that m_i is not less than R_i but less than R_{i+1}.

Step 1: DC selects a security parameter l and uses $\mathrm{gen}(l)$ to compute $(h, P_0, G_1, G_2, \hat{e})$, where P_0 is the generator of G_1 and h is the order of G_1. Then, DC selects two independent large prime numbers p and q, where $|p| = |q| = l$. Then, DC computes both public key $(N = pq, g)$ and private key (λ, μ) of the Paillier cryptosystem.

Step 2: DC computes a group of coefficient $\{a_1, a_2, \ldots, a_k\}$, where $a_1 = 1$, $a_i > \sum_{j=1}^{i-1} a_j \cdot (R_{j+1} - R_j) \cdot n$, $(i = 2, \ldots, k)$.

Step 3: DC computes another group of coefficient $\{b_1, b_2, \ldots, b_k\}$, and $b_i > b_0 + b_0 \cdot n^{i-1}$, $b_0 > a_k \cdot E \cdot n + \sum_{j=1}^{k-1} a_j \cdot R_{j+1} \cdot n$, $(i = 1, 2, \ldots, n)$, where n is the total number of users in the aggregation area.

Step 4: DC selects two secure hash functions: $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_N^*$, $H_2 : G_2 \rightarrow \mathbb{Z}_q^*$, and publishes

$$\left\{\left(R_1, g^{a_1}, g^{b_1}\right), \ldots, \left(R_k, g^{a_k}, g^{b_k}\right), H_1, H_2, N, g, P_0\right\}.$$

B. Individual Data Encryption

Assuming there are $\{user_1, user_2, \ldots, user_n\}$ in a residential area, and data m_i is collected from $user_i$. If $m_i \in [R_j, R_{j+1})$, $user_i$ is classified into subset U_j. Notably, $U = \{U_1 \cup U_2 \cup \ldots \cup U_k\}$ and $U_1 \cap U_2 = \varnothing$. Using the blinding factor k'_i and (g^{a_j}, g^{b_j}), $user_i$ encrypts its $\Delta m_i = m_i - R_j$ by computing

$$c_i = g^{a_j \cdot \Delta m_i} \cdot g^{b_j} \cdot H_1(t)^{k'_i} \bmod N^2$$

SM then sends ciphertext c_i to AG.

The blinding factor k'_i is updated as follows: all users execute the assignment operation $k'_i := N$ at first (the symbol $:=$ represents the former variable is set as the latter number). Then, $user_i$ selects random number $sk_i \in \mathbb{Z}_p^*$, and generates public key $pk_i = sk_i \cdot P_0$. The one-time key is $r_i \cdot P_0$, where r_i is a random number. $user_i$ chooses some collaborating users (maybe one or multiple people, normally three or more for security) to update the blinding factor. For example, $user_i$ chooses $user_j$ to update the blinding factor, $user_j$ selects random number $sk_j \in \mathbb{Z}_p^*$, and calculates public key $pk_j = sk_j \cdot P_0$. The one-time key is $r_j \cdot P_0$, where r_j is a random number. Then, $user_i$ sends $\{pk_i, r_i \cdot P_0\}$ to $user_j$, and $user_j$ sends $\{pk_j, r_j \cdot P_0\}$ to $user_i$.

If $i > j$, $user_i$'s k'_i is

$$k'_i := k'_i - H_2\left(\hat{e}\left(pk_j, sk_i \cdot r_i \cdot r_j \cdot P_0\right)\right)$$

and $user_j$'s k'_j is

$$k'_j := k'_j + H_2\left(\hat{e}\left(pk_i, sk_j \cdot r_j \cdot r_i \cdot P_0\right)\right)$$

Otherwise, $user_i$'s k'_i is

$$k'_i := k'_i + H_2\left(\hat{e}\left(pk_j, sk_i \cdot r_i \cdot r_j \cdot P_0\right)\right)$$

and $user_j$'s k'_j is

$$k'_j := k'_j - H_2\left(\hat{e}\left(pk_i, sk_j \cdot r_j \cdot r_i \cdot P_0\right)\right)$$

Each user can randomly select when to update the blinding factor.

C. Cipher Aggregation

AG aggregates the received ciphertext c_i by computing

$$\begin{aligned}
C &= \prod_{i=1}^{n} c_i \\
&= \prod_{i=1}^{n} g^{a_j \cdot \Delta m_i} \cdot g^{b_j} \cdot H_1(t)^{k'_i} \bmod N^2 \\
&= g^{\sum_{j=1}^{k} a_j \cdot \sum_{i \in U_j} \Delta m_i} \cdot g^{\sum_{j=1}^{k} b_j \cdot |U_j|} \cdot H_1(t)^{\sum_{i=1}^{n} k'_i} \bmod N^2 \\
&= g^{\sum_{j=1}^{k} a_j \cdot \sum_{i \in U_j} \Delta m_i} \cdot g^{\sum_{j=1}^{k} b_j \cdot |U_j|} \cdot H_1(t)^{\Omega N} \bmod N^2
\end{aligned}$$

Algorithm 1: Recover (D).

```
for $i = k$ to 1 do
    $|U_i| = (D - D \bmod b_i)/b_i$
    $D = D - (b_i \cdot |U_i|)$
end for
for $i = k$ to 1 do
    $M = (D - D \bmod a_i)/a_i$
    $D = D - (a_i \cdot M)$
    $M_i = M + R_i \cdot |U_i|$
end for
return $\{|U_1|, |U_2|, \ldots, |U_k|\}$, $\{M_1, M_2, \ldots, M_{k}\}$
```

In $\sum_{i=1}^{n} k'_i = \Omega \cdot N, \Omega \in n$ and n denote the total number of users in a residential area.

AG sends the ciphertext aggregation C to DC.

D. Aggregated Cipher Decryption

DC decrypts using the private key (λ, μ):

$$D = \left(\sum_{j=1}^{k} a_j \sum_{i \in U_j} \Delta m_i\right) + \left(\sum_{j=1}^{k} b_j \cdot |U_j|\right)$$

DC executes Algorithm 1 to obtain the number of users and the sum in each subset, $\{|U_1|, |U_2|, \ldots, |U_k|\}$, $\{M_1, M_2, \ldots, M_k\}$, and $|U_i|$ means the number of users whose data fall in U_i, and M_i is the whole power consumption in U_i. Accordingly, $\sum_{i=1}^{k} |U_i|$ is the total number of users, and $\sum_{i=1}^{k} M_i$ is the whole value in the area.

E. Fault Tolerance

As SM has limited resources, the probability of device malfunction or battery exhaustion is non-negligible. Thus, fault tolerance is essential. Specifically, AG sets the counter $count = 0$ at the beginning, and $count := count + 1$ when AG receives a ciphertext from a user, and stores the serial number of the SM.

Step 1: if $count < n$ (n is the number of users) in an aggregation period, this implies some SM(s) has/have failed. For example, if ciphertext of $user_i$ is not received, then AG publishes the message { user$_i$ break down }.

Let us assume that $user_i$ renews the blinding factor with $user_j$, and $user_j$ executes the following operations to update the blinding factor:

Step 2: $user_j$ removes $H_2(\hat{e}(pk_i, sk_j \cdot r_j \cdot r_i \cdot P_0))$ from the current blinding factor k'_i (when the user performs a blinding factor update with another user, the mutually negotiated number and the corresponding user serial number are stored in their own database), that is,

$$\begin{aligned}
k'_j := k'_j &+ H_2(\hat{e}(pk_i, sk_j \cdot r_j \cdot r_i \cdot P_0)) \\
&- H_2(\hat{e}(pk_i, sk_j \cdot r_j \cdot r_i \cdot P_0))
\end{aligned}$$

Step 3: All users upload the ciphertext again, with the exception of $user_i$.

The number of users and the relationship that need to be exchanged for the blinding factor renewal are uncertain. In addition, nobody knows who has been chosen to collaborate to update the blinding factor except the user. When the user (whose SM has failed) is announced by the AG, the bilinear pair generated with the user (whose SM has failed) is then removed.

Remark: When a new user joins the system, the new user selects some users as its collaborators, and updates its blinding factor. When an old user log out, the old user sends a logout message to all users and AG. Then, its collaborator updates the blinding factor.

V. System Evaluation

To demonstrate that our scheme achieves the design goals, we will use the following: the game between a simulator $\mathscr{C}$ and an adversary $\mathscr{A}$. In addition, $\mathscr{A}$ can execute the following queries.

1) System parameters initialization: The initialization algorithm **Init** is run by $\mathscr{C}$, which takes the security parameter k as the input and produces system parameters **Params** as the output. Then, $\mathscr{C}$ returns Params to $\mathscr{A}$.
2) Data (m_i) encryption: $\mathscr{C}$ runs the blinding factor update algorithm **Update** to generate the blinding factor k'_i. $\mathscr{C}$ also runs the encryption algorithm **Encrypt** to encrypt m_i of $user_i$ to produce ciphertext c_i with k'_i and **Params**. Then, $\mathscr{C}$ sends c_i to $\mathscr{A}$.
3) Ciphertext (c_i) aggregation: $\mathscr{C}$ runs the aggregation algorithm **Aggregate** to take ciphertext c_i as the input, and produces aggregated ciphertext C as the output. Then, $\mathscr{C}$ sends C to $\mathscr{A}$.
4) Aggregated ciphertext (C) decryption: The decryption algorithm **Decrypt** is executed by $\mathscr{C}$. **Decrypt** takes both C and **Params** as the input, in order to output the aggregated result D.
5) Fault tolerance mechanism: This is an optional mechanism, in the sense that it is not executed every time. $\mathscr{C}$ runs the fault tolerance algorithm **Tolerance** to take as input the number of received ciphertext. Then, it will produce as output the message indicating whether any user has failed. Then, $\mathscr{C}$ sends this message to $\mathscr{A}$.

Definition 1 (Correctness): The correctness of aggregation is achieved when

$$\Pr[Params \leftarrow Init(k); k'_i \leftarrow Update;$$
$$c_i \leftarrow Encrypt\left(k'_i, Params\right); C \leftarrow Aggregate\left(\{c_i\}_{i\in\Omega}\right);$$
$$Decrypt(C, Params) = 1] = 1$$

$\Omega \in n$ and n denote the total number of users in a residential area.

A. Security Theorem

Theorem 1: The proposed FTMA achieves correctness.

Proof: If $\Omega = n$ (no user has failed), each $user_i$ selects β ($\beta = 3, 4, \ldots, n$; n is the total number of users in a residential area) users to update his/her blinding factor k'_i. $user_i$ executes **Encrypt** with k'_i and **Params** to produce

$$c_i = g^{a_j \cdot \Delta m_i} \cdot g^{b_j} \cdot H_1(t)^{k'_i} \bmod N^2$$

and sends c_i to AG. AG runs **Aggregate** to generate the aggregated ciphertext C and sends C to DC. Note that the sum of a pair of blinding factors generated by the blinding factor updating process with only two participants is $2N$, due to the following reason:

The blinding factor k'_i of each $user_i$ is initialized to N at first, and assume that $user_i$, $(i \in [1, n])$ executes blinding factor update operation with $user_m$, $(m \in [1, n], i > m)$; the operations are as follows:

$$\begin{aligned} k'_i &:= k'_i - H_2\left(\hat{e}\left(pk_m, sk_i \cdot r_i \cdot r_m \cdot P_0\right)\right) \\ &:= N - H_2\left(\hat{e}(P_0, P_0)^{sk_m \cdot sk_i \cdot r_i \cdot r_m}\right) \end{aligned}$$

$user_m$ updates its blinding factor k'_m as follows:

$$\begin{aligned} k'_m &:= k'_m - H_2\left(\hat{e}\left(pk_i, sk_m \cdot r_i \cdot r_m \cdot P_0\right)\right) \\ &:= N - H_2\left(\hat{e}(P_0, P_0)^{sk_i \cdot sk_m \cdot r_i \cdot r_m}\right) \end{aligned}$$

Obviously, $k'_i + k'_m = 2N$, and the following equation is obtained:

$$\begin{aligned} C &= \prod_{i=1}^{n} c_i \\ &= \prod_{i=1}^{n} g^{a_j \cdot \Delta m_i} \cdot g^{b_j} \cdot H_1(t)^{k'_i} \bmod N^2 \\ &= g^{\sum_{j=1}^{k} a_j \cdot \sum_{i\in U_j} \Delta m_i} \cdot g^{\sum_{j=1}^{k} b_j \cdot |U_j|} \cdot H_1(t)^{\sum_{i=1}^{n} k'_i} \bmod N^2 \\ &= g^{D} \cdot \left(H_1(t)^{\Omega}\right)^{N} \bmod N^2 \\ &= g^{D} \cdot r^{N} \bmod N^2 \end{aligned}$$

where $D = (\sum_{j=1}^{k} a_j \sum_{i\in U_j} \Delta m_i) + (\sum_{j=1}^{k} b_j \cdot |U_j|)$, and $r = H_1(t)^{\Omega}$.

At this point, C satisfies the standard encryption formula of the Paillier encryption algorithm. Hence, DC obtains the aggregated result D by executing **Decrypt** with C and **Params**. In addition, Our FTMA achieves

$$\Pr[Params \leftarrow Init(k); k'_i \leftarrow Update;$$
$$c_i \leftarrow Encrypt\left(k'_i, Params\right); C \leftarrow Aggregate\left(\{c_i\}_{i\in\Omega}\right);$$
$$Decrypt(C, Params) = 1] = 1.$$

Therefore, correctness is demonstrated. ■

B. Analysis

Scenario 1: FTMA is resilient to attacks by AG.

Proof: Assuming that AG attempts to deduce a user's plaintext from ciphertext $c_i = g^{a_j \cdot \Delta m_i} \cdot g^{b_j} \cdot H_1(t)^{k'_i} \bmod N^2$.

Since AG knows nothing about the private key (λ, μ) of the Paillier cryptosystem and the blinding factor k_i', AG cannot decrypt $c_i = g^{a_j \cdot \Delta m_i} \cdot g^{b_j} \cdot H_1(t)^{k_i'} \bmod N^2$ to obtain the plaintext. ■

Scenario 2: FTMA is resilient to privacy attacks by DC.

Proof: Although DC decrypts $C = \prod_{i=1}^{n} c_i$ with the private key (λ, μ) to obtain the aggregated result $D = (\sum_{j=1}^{k} a_j \sum_{i \in U_j} \Delta m_i) + (\sum_{j=1}^{k} b_j \cdot |U_j|)$, DC cannot decrypt each single user's ciphertext $c_i = g^{a_j \cdot \Delta m_i} \cdot g^{b_j} \cdot H_1(t)^{k_i'} \bmod N^2$ without the blinding factor k_i'. Therefore, each single's plaintext is still privacy for DC. ■

Scenario 3: FTMA is resilient to collusion attack from AG and DC.

Proof: Assuming that AG and DC collude and share $c_i = g^{a_j \cdot \Delta m_i} \cdot g^{b_j} \cdot H_1(t)^{k_i'} \bmod N^2$ and (λ, μ). However, the decryption requires the blinding factor $k_i' {:=} k_i' - H_2(\hat{e}(pk_j, sk_i \cdot r_i \cdot r_j \cdot P_0))$ ($user_i$ updates the blinding factor with $user_j$). Even if DC and AG obtain $(pk_i, r_i \cdot P_0, pk_j, r_j \cdot P_0)$ from the public channel, it is computationally challenging for them to compute $(sk_i \cdot r_i \cdot r_j \cdot P_0)$ from $pk_i = sk_i \cdot P_0$ and $r_i \cdot P_0, r_j \cdot P_0$ due to the hard problem assumption CDHA. In addition, sk_i and r_i can not be deduced from pk_i and $r_i \cdot P_0$ due to the discrete logarithm hard problem over elliptic curve. Thus, k_i' is unknown by DC and AG. ■

Scenario 4: Even if several users collaborate and attempt to obtain a user's data, this collusion attack is computationally infeasible.

Proof: Assuming that $user_i$ ($i \in [1, n]$, n is the number of users) executes blinding factor update with $user_m$ ($m \in [1, n]$), then executes with $user_k$, ($k \in [1, n]$) (i.e., the number of collaborators is 2), where $i > m$ and $i > k$. The process of blinding factor update is as follows:

$user_i$'s k_i' is initialized to N, $user_i$ updates the blinding factor k_i' with $user_m$ as

$$k_i' := k_i' - H_2\left(\hat{e}\left(pk_m, sk_i \cdot r_i \cdot r_m \cdot P_0\right)\right).$$

Then, $user_i$ executes the update with $user_k$ as

$$k_i' {:=} k_i' - H_2\left(\hat{e}(pk_m, sk_i \cdot r_i \cdot r_m \cdot P_0)\right) - H_2\left(\hat{e}(pk_k, sk_i \cdot r_i \cdot r_k \cdot P_0)\right).$$

■

Assume $user_m$ and $user_k$ try to collude with DC to obtain m_i of $user_i$. Under the assumption that attackers know m_i falls in $[R_j, R_{j+1})$, they can recover m_i by obtaining Δm_i where $\Delta m_i = m_i - R_j$. To obtain Δm_i from ciphertext $c_i = g^{a_j \cdot \Delta m_i} \cdot g^{b_j} \cdot H_1(t)^{k_i'} \bmod N^2$, the privacy key (λ, μ) and the blinding factor k_i' are necessary. For recovering the blinding factor k_i', $user_m$ provides the number negotiated with $user_i$, that is, $H_2(\hat{e}(pk_m, sk_i \cdot r_i \cdot r_m \cdot P_0))$. Let us also assume that $user_k$ provides the negotiated number $H_2(\hat{e}(pk_k, sk_i \cdot r_i \cdot r_k \cdot P_0))$, which is negotiated with $user_i$ to DC. Then, DC provides the private key (λ, μ) of the Paillier cryptosystem and utilizes k_i' to derive Δm_i from ciphertext $c_i = g^{a_j \cdot \Delta m_i} \cdot g^{b_j} \cdot H_1(t)^{k_i'} \bmod N^2$. Although the attacker has the private key, the user's m_i cannot be decrypted unless the attacker also has access to the correct blind factor k_i'. This is because the victim may update the blinding factor once more before the attacker(s) obtain the latest correct blinding factor, regardless of whether the attacker has updated the blinding factor with the victim. Hence, we demonstrate that the scheme is resilient to collusion attack.

TABLE I
COMPUTATIONAL COSTS: A COMPARATIVE SUMMARY

	NPSA[23]	PPMA[25]	PPSO[28]	Our scheme
AG	$(n \cdot T_{mul})$	$(n \cdot T_{mul})$	$(n \cdot T_{mul})$	$(n \cdot T_{mul})$
user	$2T_e$+$2T_{mul}$+T_H	$5T_e$+$2T_{mul}$+T_H	$3T_e$+T_{mul}+T_H	$4T_e$+$8T_{mul}$+ $4T_H + 3T_b$

Scenario 5: Fault tolerance is achieved, and the troubleshooting operation does not leak anything useful about the users' collaboration relation.

Proof: Here, fault tolerance refers to terminal devices failing to send data normally. Cases such as AG failure and deliberate sending of fake data by SM are beyond the scope of fault tolerance. ■

The fault-tolerant mechanism typically includes fault detection, troubleshooting, and aggregation operations. We implement these in three steps, namely: 1) using a counter to detect the fault; 2) when the collaborator of the faulty equipment updates the blind factor, other entities are unaware; and 3) when some users in the system fail, the system can still obtain their aggregated value by making normal users reupload the ciphertext.

In other words, FTMA can still be functionally executed with our fault tolerance mechanism to obtain the result even if some SMs fail to work. Meantime, the aggregated result excluding the data of the failed SM is almost identical to the original aggregation value for the data consumer out of the aggregation system. Furthermore, since all devices reupload the data, there is no way to know who had ever collaborated with the failed device, thus protecting the privacy of the user's blind factor. Although this process incurs additional computational and communication costs, the probability of executing the fault tolerance mechanism is very low. In other words, we achieve an optimal trade-off between computation and communication overheads and fault tolerance, and the additional overheads will be presented next.

C. Performance Evaluation

In this section, the performance comparative summary of FTMA and NPSA [23], PPMA [25], and PPSO [28] is presented.

1) Computation Cost: The execution times of exponentiation, point multiplication, hash and bilinear pairing T_e, T_{mul}, T_H and T_e are, respectively, 0.572, 1.476, 1.001, and 2.187 ms, from the executions on a laptop with Intel Core i5-3230 M CPU @2.60 GHz and 4.00 GB memory, using the PBC and OpenSSL library.

1) User's computation cost: In FTMA, the cost incurred by the user during encryption is $4T_e + 2T_{mul} + T_H$, and the blind factor update incurs a computational cost of $3 \times (2T_{\text{mul}} + T_b + T_H)$; hence, totaling $4T_e + 8T_{mul} + 4T_H + 3T_b$.
2) AG's computation cost: In the data aggregation process, four schemes (i.e., our FTMA, NPSA [23], PPMA [25], and PPSO [28]) incur cost of $(n \cdot T_{mul})$.

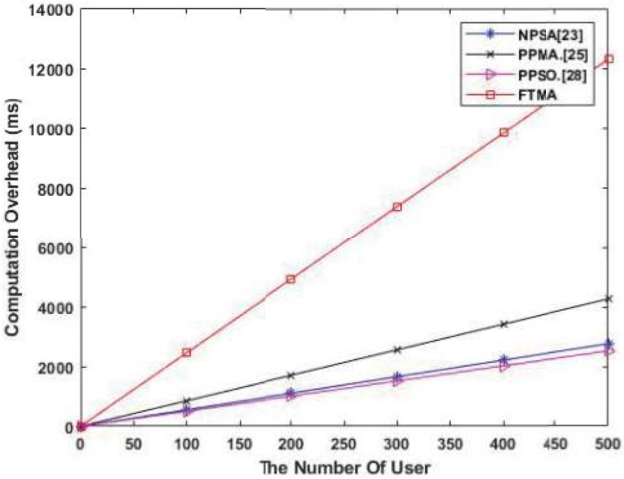

Fig. 3. Computation costs: a comparative summary.

TABLE II
SECURITY FEATURES: A COMPARATIVE SUMMARY

	NPSA[23]	PPMA[25]	PPSO[28]	Our scheme
TTP required	Yes	Yes	No	No
Fault Tolerance	No	No	Yes	Yes
Robustness	Low	Low	Low	High
Man-in-the-middle Attack	Yes	Yes	Yes	Yes
Collusion Attack	No	No	No	Yes
Dynamic	No	Yes	Yes	Yes

A comparative summary is shown in Table I. Fig. 3 depicts the computation overheads of FTMA, NPSA [23], PPMA [25], and PPSO [28], with varying user group size. A comparative summary of the properties is presented in Table II, and FTMA is more robust than other schemes, with negligible additional overheads.

As the blinding factor is key to protecting user-sensitive data, it can have an impact on user's computation and communication costs. Thus, we evaluated the impact of the blinding factors generated by different numbers of users on the system runtime. We assume that there is an aggregation area with 1000 users. Fig. 4 shows the computation overheads incurred by the updates of the blinding factor with varying sizes. When the blinding factor is generated by 10 users, the computation overhead is 43 ms, and when the blinding factor is generated by 500 users, the computation overhead is 117 ms. This demonstrates the efficiency of our protocol.

2) Dynamic Customer's Enrollment and Revocation: Assuming that 100 users join or leave a residential area with 1000 users, and eight subsets in data aggregation.

1) When a new user joins the system, the additional operation cost is $6 \times (2T_{\text{mul}} + T_b + T_H)$ (and this is not affected by the number of users in the system).
2) When a user exits from the system, the operations include only three subtraction or addition (i.e., the cost is negligible).

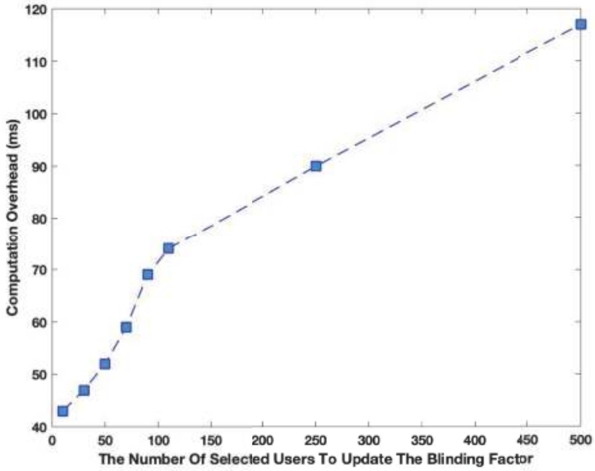

Fig. 4. Costs incurred for blinding factor updating.

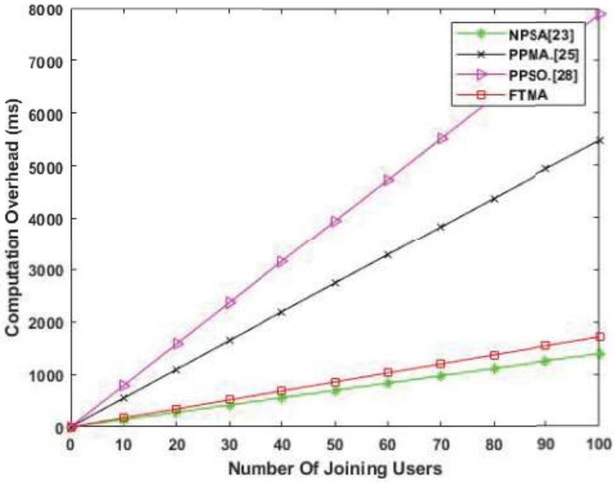

Fig. 5. Computation costs incurred by user(s) joining: a comparative summary.

Compared with PPMA [25], PPSO [28], and NPSA [23] in Figs. 5 and 6, FTMA is efficient and practical.

3) Fault Tolerance: If the SM collects and sends data every 15 min, the annual computation and communication overheads are 35 040. If we assume an SM fails once in 3 years, the failure probability of each computation and communication is nearly $1/100\,000 = 0.00001$. Accordingly, the probability that at least one user fails to work in an residential area with 100 users is $1 - 0.99999^{100} \approx 0.001$. Since the failure probability of standard industrial SM is much lower than 0.00001, the probability of executing the fault tolerance mechanism is reduced to nearly zero. Under the assumption that the length of Paillier encryption output is 256 bits, when the fault tolerance mechanism is executed, each SM's additional computation cost is only $4T_e + 2T_{\text{mul}} + T_H$. The AG's additional cost is $((n-1) \times T_{\text{mul}})$, where n is the total number of the aggregation

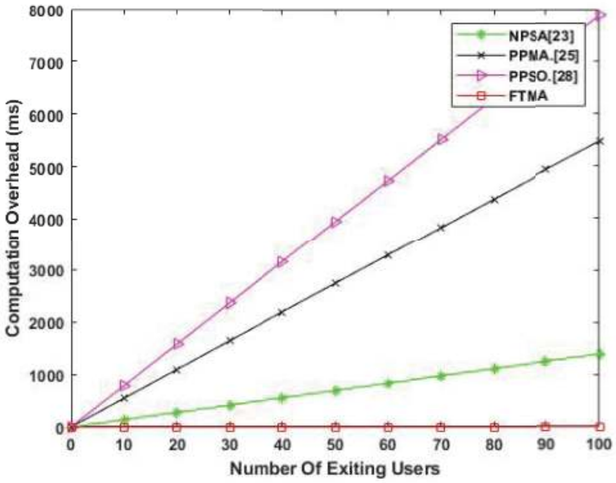

Fig. 6. Computation costs incurred by user(s) exiting: a comparative summary.

area. The SM's additional communication cost is 256 bits and AG's additional communication cost is $(n-1) \times 256$ bits.

VI. Conclusion

In this article, a data aggregation scheme without relying on TTP was proposed to achieve multisubset aggregation. Using our scheme allows one to obtain and publish more fine-grained result. The in-built fault-tolerance mechanism also guarantees the system to be functional even if some terminal devices fail.

References

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

[2] Z. Sui and H. de Meer, "An efficient signcryption protocol for hop-by-hop data aggregations in smart grids," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 1, pp. 132–140, Jan. 2020.

[3] P. Zeng, B. Pan, K.-K. R. Choo, and H. Liu, "MMDA: Multidimensional and multidirectional data aggregation for edge computing-enhanced IoT," *J. Syst. Architecture*, vol. 106, 2020, Art. no. 101713.

[4] J. Hajny, P. Dzurenda, and L. Malina, "Privacy-enhanced data collection scheme for smart-metering," in *Proc. Int. Conf. Inf. Secur. Cryptology*, Springer, 2015, pp. 413–429.

[5] T. Jung, X.-Y. Li, and M. Wan, "Collusion-tolerable privacy-preserving sum and product calculation without secure channel," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 1, pp. 45–57, Jan.–Feb. 2015.

[6] D. He, N. Kumar, and J.-H. Lee, "Privacy-preserving data aggregation scheme against internal attackers in smart grids," *Wireless Netw.*, vol. 22, no. 2, pp. 491–502, 2016.

[7] Y. Liu, G. Liu, C. Cheng, Z. Xia, and J. Shen, "A privacy-preserving health data aggregation scheme," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 8, pp. 3852–3864, 2016.

[8] E. Vahedi, M. Bayat, M. R. Pakravan, and M. R. Aref, "A secure ECC-based privacy preserving data aggregation scheme for smart grids," *Comput. Netw.*, vol. 129, pp. 28–36, 2017.

[9] Z. Sui, M. Niedermeier, and H. de Meer, "RESA: A robust and efficient secure aggregation scheme in smart grids," in *Proc. Int. Conf. Crit. Inf. Infrastructures Secur.*, Springer, 2015, pp. 171–182.

[10] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet Things J.*, vol. 6, pp. 4755–4763, Jun. 2019.

[11] F. Borges and M. Mühlhäuser, "EPPP4SMS: Efficient privacy-preserving protocol for smart metering systems and its simulation using real-world data," *IEEE Trans. Smart Grid*, vol. 5, no. 6, pp. 2701–2708, Nov. 2014.

[12] Z. Erkin, "Private data aggregation with groups for smart grids in a dynamic setting using CRT," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2015, pp. 1–6.

[13] X. Liu, Y. Zhang, B. Wang, and H. Wang, "An anonymous data aggregation scheme for smart grid systems," *Secur. Commun. Netw.*, vol. 7, no. 3, pp. 602–610, 2014.

[14] B. Pan, P. Zeng, and K.-K. R. Choo, "An efficient data aggregation scheme in privacy-preserving smart grid communications with a high practicability," in *Proc. Conf. Complex Intell. Softw. Intensive Syst.*, Springer, 2017, pp. 677–688.

[15] H. Shen, M. Zhang, and J. Shen, "Efficient privacy-preserving cube-data aggregation scheme for smart grids," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 6, pp. 1369–1381, Jun. 2017.

[16] 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.

[17] Y. Zhang *et al.*, "Privacy-preserving data aggregation against false data injection attacks in fog computing," *Sensors*, vol. 18, no. 8, 2018, Art. no. 2659.

[18] C. Zhang, L. Zhu, C. Xu, K. Sharif, and X. Liu, "PPTDS: A privacy-preserving truth discovery scheme in crowd sensing systems," *Inf. Sci.*, vol. 484, pp. 183–196, 2019.

[19] Department of Energy and Climate Change, "The smart metering system," 2014. [Online.] Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/ attachment_data/file/426135/Smart_Metering_System_leaflet.pdf

[20] M. A. Mustafa, N. Zhang, G. Kalogridis, and Z. Fan, "DEP2SA: A decentralized efficient privacy-preserving and selective aggregation scheme in advanced metering infrastructure," *IEEE Access*, vol. 3, pp. 2828–2846, 2015.

[21] M. A. Mustafa, S. Cleemput, A. Aly, and A. Abidin, "A secure and privacy-preserving protocol for smart metering operational data collection," *IEEE Trans. Smart Grid*, vol. 10, pp. 6481–6490, Nov. 2019.

[22] A. Alsharif, M. Nabil, M. M. Mahmoud, and M. Abdallah, "EPDA: Efficient and privacy-preserving data collection and access control scheme for multi-recipient AMI networks," *IEEE Access*, vol. 7, pp. 27829–27845, 2019.

[23] R. Lu, K. Alharbi, X. Lin, and C. Huang, "A novel privacy-preserving set aggregation scheme for smart grid communications," in *Proc. IEEE Global Commun. Conf.*, 2015, pp. 1–6.

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

[25] 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.

[26] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "EDAT: Efficient data aggregation without TTP for privacy-assured smart metering," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.

[27] D. He, N. Kumar, S. Zeadally, A. Vinel, and L. T. Yang, "Efficient and privacy-preserving data aggregation scheme for smart grid against internal adversaries," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2411–2419, Sep. 2017.

[28] 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.

[29] J. Jiang *et al.*, "Anomaly detection with graph convolutional networks for insider threat and fraud detection," in *Proc. IEEE Mil. Commun. Conf.*, Norfolk, VA, USA, Nov.12–14, 2019, pp. 109–114.

[30] J. Chen and H. Wee, "Doubly spatial encryption from DBDH," *Theoret. Comput. Sci.*, vol. 543, pp. 79–89, 2014.

[31] Y.-N. Liu, Y.-P. Wang, X.-F. Wang, Z. Xia, and J.-F. Xu, "Privacy-preserving raw data collection without a trusted authority for IoT," *Comput. Netw.*, vol. 148, pp. 340–348, 2019.