# Practical Privacy-Preserving Scheme With Fault Tolerance for Smart Grids

**Chang, Yuan; Li, Jiliang; Lu, Ning; Shi, Wenbo; Su, Zhou; Meng, Weizhi**

*Published in:*
IEEE Internet of Things Journal

*Link to article, DOI:*
10.1109/JIOT.2023.3303010

*Publication date:*
2024

*Document Version*
Peer reviewed version

*Citation (APA):*
Chang, Y., Li, J., Lu, N., Shi, W., Su, Z., & Meng, W. (2024). Practical Privacy-Preserving Scheme With Fault Tolerance for Smart Grids. *IEEE Internet of Things Journal*, *11*(2), 1990 - 2005. https://doi.org/10.1109/JIOT.2023.3303010

# Practical Privacy-Preserving Scheme with Fault Tolerance for Smart Grids

Yuan Chang*, Jiliang Li*, Ning Lu†, Wenbo Shi†, and Zhou Su*, *Senior Member, IEEE,* Weizhi Meng‡
*School of Cyber Science and Engineering, Xi'an jiaotong University, Xi'an, China
†School of Computer Science and Engineering, Northeastern University, Shenyang, China
‡Department of Applied Mathematics and Computer Science , Technical University of Denmark, Copenhagen, Denmark

***Abstract*—In smart grid services, the leakage of crowdsourced consumption data on smart meters poses potential risks of privacy disclosure and data misuse. Existing solutions, which rely on complex encrypted computations, are often impractical for resource-limited smart meters due to their high computation and storage resource requirements. To address these challenges, this paper proposes a practical privacy-preserving scheme with fault tolerance for smart grid services named 3PFT. In our scheme, we employ a masking approach that ensures user privacy preservation on smart meters while consuming minimal resources. Unlike existing masking schemes, 3PFT provides fault tolerance, supports complex data analysis tasks, and mitigates vulnerabilities to key leakage attacks. To achieve these objectives, we incorporate a secret sharing technique into the masking approach, enabling the recovery of the master key using only a portion of the data. Additionally, we design a flexible data aggregation protocol for 3PFT, facilitating the execution of diverse data analysis missions such as load forecasting in smart grids. Furthermore, we introduce a negotiation-based key update method to enhance the protocol's forward security and alleviate the additional overhead on smart meters. Lastly, we provide a rigorous proof of privacy preservation and fault tolerance for our scheme and validate its feasibility and effectiveness through extensive simulations.**

***Index Terms*—Smart grid, Privacy-preserving, Load prediction, Fault tolerance**

## I. Introduction

With the exponential growth of power data and the advancement of smart grid technology, crowdsourcing has revolutionized the generation and utilization of power grid data [1]. The analysis results obtained from gathering extensive data on the power grid can aid in task scheduling, control and management of indicators, as well as prompt problem diagnosis. For instance, smart meters ($SM$) installed in buildings collect consumption data every 15 minutes and transmit it to energy suppliers ($ES$), who can use it to predict future electricity usage trends based on consumption [2]. Furthermore, the decentralized electrical appliances' crowdsourced data can help formulate electricity prices based on demand-response mechanisms. However, the use of crowdsourcing mechanisms raises concerns regarding the security and privacy of users. Data leakage during the data collection process can potentially compromise the privacy of home users. For example, malicious attackers can use information on low power consumption to deduce that only children are present in the house, leading to potential theft or fraud.

Existing privacy-preserving solutions can be classified into two categories: plaintext-based privacy schemes and ciphertext-based cryptography operations [3]–[11]. Plaintext-based schemes aim to protect privacy by breaking the link between plaintext and the user's identity through anonymization or data load balancing. However, these schemes are vulnerable to background knowledge attacks, which suggests that an attacker can breach privacy protection by mining user-related identity data. Ciphertext-based schemes perform cryptographic operations on consumption data to ensure strong security but have low practicality. For example, symmetric encryption algorithms offer no homomorphism capability, which reduces the efficiency of data collection. The semi-homomorphic encryption algorithm uses additional high computational operations, such as modular multiplication and modular exponentiation. Table 1 illustrates the computational dilemma faced by current smart meters ($SMs$), which can only support a few lightweight operations, such as hash, AES, and XOR.

Recently, a promising masking approach has been proposed that aims to achieve a balance between security and performance, and it is potentially deployable in $SMs$. This approach involves adding a secure random noise value $k \mod n$ to the user's data to obtain a masked value $m'$. Here, $k$ is in the range $[1, n]$, and $n$ is a large random number. In this scheme, each $SM_i$ uses a unique masked value $k_i$ to encrypt consumption data, and $ES$ uses the master key $\sum k_i$ to decrypt the aggregated data. Despite the practicality associated with current masking schemes [12], [13], several significant challenges still exist.

TABLE I: Time cost of different operations between computer and SM

| Operation | PC[1] | Smart Meter[2] |
|---|---|---|
| SHA256 | $0.027ms$ | $2.16ms$ |
| Add | $0.018ms$ | $1.44ms$ |
| Multiplication | $0.97ms$ | $0.77s$ |
| Exponentiation | $1.39ms$ | $1.11s$ |
| AES256 | $0.014ms$ | $1.12ms$ |

[1] PC configuration: Intel 3.0GHz i5-8500 CPU
[2] SM configuration: 50HZ DDZY110C-Z CPU

Jiliang Li is the corresponding author (e-mail: jiliangli@foxmail.com)

In terms of fault tolerance, the proposed scheme should offer high error tolerance, which is crucial to handle a large amount of lost data due to attacks in the smart grid. The key relation $k = \sum k_i$ in masking-based schemes can cause aggregation decryption to fail, thus, making incorrect individual data an issue. Data loss and forgery are common in the data transmission of the smart grid, and some proposed schemes suggest constructing a ring network or challenge-response mode to determine a disconnected $SM$ during the transmission [14]. However, changing network structures and online detection of smart meters can lead to additional communication overhead and maintenance costs. Hence, we propose an adaptive fault tolerance mechanism that minimizes the extra overhead.

Flexibility is another crucial feature of our proposed scheme. We define flexibility as that the entity (such as an electric power company) can carry out flexible and granular data analysis based on the aggregated data. The granularity of the aggregation must be adaptive to the needs of the entity, and the scheme should provide flexible data analysis and aggregation to meet the requirements of various entities. For instance, franchise stores require consumption information of their stores to analyze costs, and refrigerator manufacturers need refrigerator consumption data of residents in a region to adjust their market strategy [15]. Moreover, the key data should be hidden in the aggregate data that is submitted; this poses difficulties for data analysis. Basic statistical analysis requires calculations such as mean, extreme, and variance, which are challenging to compute in lightweight masking schemes [16].

Another critical feature of our scheme is forward security. The masking method relies on the randomness of the blinding factor $k_i$ for its security, which decreases with the increasing number of ciphertexts. The encryption upper bound $N$ limits masking's security, and performing more than $N$ encryptions can compromise historical data; masking may no longer be secure beyond this limit. Consequently, we propose a masking-supported key update approach to ensure forward security and consider the cost of applying it to $SM$.

To finely resolve the above three challenging issues on masking schemes, we propose a practical privacy-preserving scheme with fault tolerance for smart grids (named 3PFT), and the contributions is as below.

- Firstly, in 3PFT, we use Shamir's threshold secret sharing technique to generate masking keys (i.e. shares) and recover the master key, which can achieve high error tolerance.
- Further, we design a fine-grained data aggregation protocol on the basis of 3PFT (DA-3PFT) to support short-term and long-term load prediction. Auxiliary values such as mean, variance, and outliers can support fine-grained prediction of the time series method. At the same time, HMAC and RSA methods are used for lightweight authentication to protect the integrity of data transmission.
- Besides, a negotiation-based key update method can be deployed in DA-3PFT to improve security. Two users modify the key adaptively without affecting the decryption correctness of the masking. The user's personal device (e.g., cell phone), rather than the smart meter, bears most of the communication and computing costs.
- Lastly, security analysis and performance analysis prove that our scheme is secure and lightweight.

In Section 2, we describe the related work. In Section 3, the proposed 3PFT is introduced. Section 4 presents the DA-3PFT aggregation protocol, and Section 5 presents the key update method. Then, we will describe the security analysis and performance evaluation in Section 6 and Section 7. The last section concludes this paper.

## II. Relevant Background and Literature

### A. Background

In a standard aggregation scenario, the decryption key of the aggregated data is related to the encryption data key, which can be formalized as an additive homomorphic property. Specifically,

$$\sum_{i=0}^{n} En_{k_i}(m_i) = En_k(\sum_{i=0}^{n} m_i), \tag{1}$$

where $k_i$ represents the subkey, $k$ denotes the aggregate key, and $k$ value relies on the subkey $k_i$. However, in the smart grid setting, it often happens that $k_i$ cannot be aggregated, leading to $k$ value changes for each aggregation. We define fault tolerance as the following formula:

$$\sum_{i\in U} En_{k_i}(m_i) = En_k(\sum_{i\in U} m_i), \tag{2}$$

where $U$ represents the set of each upload. The challenge with fault tolerance is that the set of uploads is not fixed, making the above equality hold not all the time. Existing fault-tolerant schemes for data aggregation can be broadly classified into the following two types.

**Active Detection** In this approach, the gateway or control center detects each uploaded meter set through active detection or verification and adjusts the key accordingly. This can be achieved by supplementing the lost key $k_i$ (for $i \notin U$) or modifying the key $k$. This method ensures accurate collection of the aggregated dataset but incurs a significant communication overhead (including the detection and modification cost) [17]–[20].

**Passive Discard** This approach involves the control center never detecting which meters have failed. It collects $t$ (where $t < n$) data at a time achieved by embedding secret sharing into encryption. $t$ represents the threshold of correct aggregation. This strategy incurs no additional communication overhead but may lead to inaccurate data, as some data is abandoned [21], [22].

Therefore, to address these challenges, this paper implements a secret sharing scheme under the masking method based on the second method, starting from lightweight aggregation. The mean and variance are also attached to improve the accuracy of data analysis.

### B. Related Work

Numerous privacy-preserving solutions have been proposed for smart grids, which are divided into four categories depending on how the data is processed (1) Homomorphic encryption, (2) Masking, (3) Anonymization, and (4) Load Mitigation. We will discuss each of the four aggregation schemes in detail below. In addition, we also highlight certain current challenges related to fault tolerance and flexibility.

**Homomorphic encryption** This scheme leverages public-key cryptography to encrypt data for privacy protection. Shen et al. proposed a data aggregation scheme based on Paillier encryption, which sends user's multi-dimensional data in the form of polynomials. In the aggregation, different coefficients ensure that the data of each dimension is independently aggregated [7]. Liu Yi et al. proposed a trusted third-party-free aggregation protocol using the EC-ElGamal cryptosystem [23]. Liu et al. proposed an aggregation scheme based on fog computing, which uses a double Trapdoor Decryption cryptosystem to broaden the analysis results and support users and energy providers to obtain data independently [11]. Amin et al. proposed a novel homomorphic privacy-preserving protocol (NHP3) based on Nyberg's accumulator, enabling multi-category aggregation and batch verification on the aggregator [18]. However, the ciphertext expansion and efficiency problems make homomorphic encryption challenging to apply in smart grids.

**Masking**. This scheme uses random numbers to blind the data, which is more lightweight. Gope et al. proposed a privacy-protected data aggregation protocol. To protect the privacy of the smart meter, each user is assigned multiple pseudonyms to correspond to the data sent each time, which can ensure the anonymity of the data [12]. Song et al. [24] proposed a dynamic data aggregation protocol, where two users can modify the masking parameters simultaneously through interaction. The temporal and spatial aggregation protocol, proposed by Knirsch et al. [13], constructs a ring and star network, and the smart meter can judge the condition of the neighboring smart meter by sending signals. Building on this idea, Gope et al. [25] proposed a lightweight protocol that satisfies spatial aggregation. Su et al. [26] proposed the lightweight and communication-efficient data aggregation (LCEDA) scheme to reduce additional communication costs. However, these masking protocols do not support fault tolerance and fine-grained analysis.

**Anonymization**. This scheme does not process the data but hides the user's identity to disconnect the data from the owner. In 2010, Costa proposed an anonymous electricity meter to protect identity privacy [3]. The utility company can only see the low-frequency ID (IFID) in procurement and deployment. The high-frequency ID (HFID) is protected, and only the manufacturer and third-party hosting service know it. The connection between the two IDs is disconnected and saved in the smart meter as a configuration file. Mohammad proposed a scheme based on an onion network to meet source anonymity, destination anonymity, and routing anonymity [4]. At the same time, the use of pseudo-IDs further facilitates the anonymity of real identities. The above scheme can achieve weak privacy protection through identity privacy protection. However, anonymity contradicts some businesses of the smart grid. Implementing billing and demand responses is hard without knowing the user's identity.

**Mitigation**. The main idea of the scheme is to moderate the consumption data of the smart meter. Varodayan proved that rechargeable batteries could effectively reduce the rate of information leakage [5]. The battery can be charged and discharged according to the current data to stabilize the load for some time. This scheme depends on the configuration of the battery, costs more, and is not practical. In addition, adding a specific distribution of noise (e.g., Gaussian, Laplacian) to the electricity consumption data can achieve good privacy protection. In particular, there is a difference between differential privacy and masking, and the blinding factor of masking is randomly selected, that is, uniformly distributed. For differential privacy, the added noise is used as a part of the data, while masking, as an encryption scheme, can remove the blinding factor and decrypt data [27]–[30]. The research shows that the data with noise is challenging to analyze by machine learning [31][1]

## C. Relevant Solutions to The Challenges

To address the challenges presented by the introduction, some attempts have been made in the latest related literature, and we present them.

**Fault Tolerant.** To achieve fault tolerance, existing schemes can be divided into two ideas: (1) ignore errors and (2) detect errors and resolve them.

Ahsan et al. combined the secret sharing scheme with the paillier encryption scheme to achieve fault tolerance [22]. The proposed scheme can resist false data injection attacks by filtering out the inserted values from external attackers. Wu et al. propose homomorphic signatures combined with secret sharing to detect aggregator errors so that anyone can verify that the result has been correctly computed [21]. Amin et al. use a group verifier to quickly check meter reports and classify consumption data for multidimensional aggregation [18]. However, the above three schemes bring heavy computational overhead to $SM$ while achieving powerful functions, which is not conducive to applying $SM$ in the current power grid.

Wang et al. achieve fault tolerance by actively detecting the number of erroneous $SM$ and adjusting blinding privacy, and at the same time, can obtain the total power consumption of users in each region to achieve multi-subset aggregation [17]. Xu et al. proposed additional request-response interactions generated by fog nodes and $SM$ to cope with $SM$ that failed to submit data [20]. Chang et al. achieve fault tolerance for time series aggregation by providing relevant key reconstruction $k$ at each online $SM$ [19]. The above three papers solve fault tolerance but introduce more communication overhead, which is not conducive to the efficient execution of the protocol.

**Flexibility.** The existing flexibility schemes mainly extend the functionality for the availability of aggregated data. Chen et al. proposed a multi-data aggregation scheme based on dynamic membership groups. Dynamic join, dynamic leave,

[1] This paper only illustrates the challenges of differential privacy. Much work is being done to leverage differential privacy-based mechanisms for privacy-preserving machine learning. But differential privacy is out of the scope of this paper.

and metering replacement techniques are proposed to realize dynamic membership [32]. Zhan et al. proposed an efficient data aggregation scheme supporting privacy protection and functional queries. The proposed scheme allows the control center and users to initiate various functional queries on encrypted data [33]. Yan et al. proposed an efficient server-oriented multi-task data aggregation scheme [34]. The proposed scheme can aggregate multiple concurrent tasks from multiple requesters. Gupta et al. developed an OBSCURE system for aggregate queries over conjunctive and disjunctive predicates for secret sharing [35]. This technique deals with secret shared data outsourced by multiple database owners and allows users to query the secret shared data. However, few papers focus on the conflict between aggregation and data analysis, and the given aggregated data often cannot realize fine-grained data analysis.

**Forward Security.** Existing schemes focus on ensuring forward security in the presence of key compromise and dishonest insiders. Wang et al. proposed a novel cryptographic scheme to ensure forward secrecy and non-repudiation [36]. The scheme is based on key derivation symmetric encryption and online/offline signature construction, which can realize efficient operation in the embedded controller. Niu et al. proposed a blockchain key aggregation searchable encryption scheme with auxiliary input [37]. The scheme is proven to prevent key exposure under the Diffie-Hellman assumption. Chen et al. proposed an improved DeepPAR scheme to solve the problem of proxy server key leakage caused by re-encryption key generation, and it can resist collusion attacks [38]. Zhang et al. proposed a key-compromise resilient encrypted data aggregation scheme with lightweight verification in smart grids [39]. The private key of the control center is leaked in time, and any adversary can not destroy the privacy of the user. The property of forward security depends on the scheme itself, and all the above schemes are implemented in the public key cryptographic scheme. Therefore, it is still a challenge to implement forward security based on masking encryption schemes to prevent key compromise.

## III. Our Proposed 3PFT

In this section, we first present the system model and adversary model, and then the components and workflow of 3PFT are introduced.

### A. System model

The system model considers a typical smart grid communication architecture, comprising a large number of smart meters ($SM$), some gateways ($GW$), a cloud server ($CS$), and a trusted key server ($KS$), as shown in Figure 1. The system entities are defined as follows:

- $KS$: The key server is responsible for distributing keys for the gateway, cloud, and $SM$. The key is used for encryption and authentication.
- $SM$: The Internet of Things device is installed in each home building and calculates the consumption of all household electrical equipment in the home area network (HAN). It encrypts the energy consumption data and uploads it to the gateway.
- $GW$: The gateway between the cloud and smart meter acts as an aggregator, accumulating all collected data ciphertexts and uploading them to the cloud.
- $CS$: The cloud server is leased by the energy supplier and is responsible for information storage, analysis, planning, and decision-making.
- $ES$: The energy supplier makes energy supply and transmission decisions favorable to the smart grid according to the analysis results of $CS$.

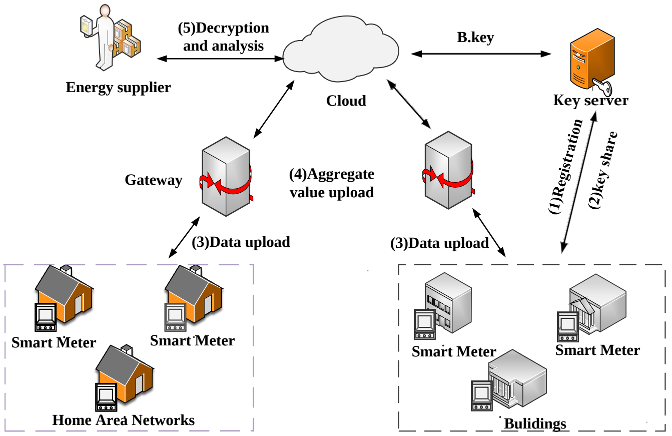


Fig. 1: System model

### B. 3PFT

The 3PFT workflow is first introduced, then we provide two crucial tools used in step (3) to achieve fault tolerance and lightweight authentication.

*1) Workflow:* In the 3PFT scheme, the workflow of the above entities is described as follows: (1) $SM$ registers with $KS$: when a new home user moves in, workers install the power company's $SM$s for each home to ensure that the data can be uploaded online. The $SM$ will legally register its ID number and user identity with $KS$ to upload data. If the ID number is correct and not registered, $KS$ will record the ID number and user identity and send it to $ES$. $ES$ will use this information to send bills and interactive information to users. (2) $KS$ sends keys to various entities: when most $SM$ have been determined, $KS$ will calculate the encryption key and signature key required for $SM$s, $GW$s, and $CS$. These keys are sent to each entity through a secure channel. $KS$ will reserve some keys to prepare for the addition of new meters in the future. (3) $SM$ encrypts and uploads data: $SM$ collects real-time consumption data of household appliances and aggregates them together. According to the consumption receipt demand of the $ES$, the $SM$ transmits the collected information at a specific time interval. In the transmission process, the confidentiality and integrity of data shall be ensured, and the transmission channel is not required to be a secure channel. The encryption and authentication components used in this step are described below. (4) $GW$ aggregates data: $GW$ receives data, determines the message source's reliability, and then aggregates the data. These data are prepared for $ES$ analysis. In addition, $ES$ needs to calculate a lot of auxiliary data and ensure data integrity. (5) $ES$ decrypts and analyzes the aggregate data: $ES$ also confirms the aggregated data source's reliability and decrypts it. Analyze aggregate data and auxiliary data. The analysis results are used for the electricity dispatching of the smart grid. The specific analysis method is analyzed in the protocol.

*2) Masking Based On Secret Sharing:* Secret sharing technology has a natural fault tolerance mechanism. Note that the threshold for secret sharing also means that it does not need to be all. Because the system model allows the trusted $KS$ entity can issue the key, the Shamir secret sharing technology can be well applied. Although this combination seems easy, we are the first to do it to the best of our knowledge. The following equation calculates the share of $k_i$ as $k$:

$$k_i = f(i) = k + a_1 \cdot i^1 + a_2 \cdot i^2 + ... + a_t \cdot i^t, \tag{3}$$

Where $t$ is the threshold for aggregation [2], and the coefficients $a_1, a_2, ..., a_t \in GF(2_q)$ of the t-degree polynomial are derived from a TRNG. When more than $t$ $k_i$ are aggregated, the key $k$ can be restored according to $k = \sum_{i=1}^{t} \lambda_i k_i$. The $\lambda_i$ is the Lagrange coefficient, and data $m_i$ can be combined with a $k_i$ to become masking data. Note that we do not mask $m_i$ directly with $k_i$ here, as this would result in the inability to restore $k$ during aggregation. So we chose to use $\lambda_i k_i$ to mask $m_i$:

$$m_i{}^* = m_i + \lambda_i k_i \pmod M, \tag{4}$$

where $M$ must be much larger than the aggregated value to ensure data security. In the aggregation process, due to the $SM's$ damage, the aggregation of some data does not affect the accuracy. The key $k$ can be used to decrypt the aggregated usage data $m_a$. $\sum_{n=1}^{t} m_i{}^* = \sum_{n=1}^{t}(m_i + \lambda_i k_i) = \sum_{n=1}^{t} m_i + k$. So $m_a = \sum_{n=1}^{t} m_i{}^* - k$. Because of masking, the data of a single user cannot be obtained by the $AG$ and $ES$. It is worth noting that only sequential $\lambda_i k_i$ can make the 3PFT scheme work. Therefore, each $SM$ must record its order, which can be the order of registration.

*3) Hash Message Authentication Code (HMAC):* The message authentication code (MAC) is required to prevent tampering with the message by the adversary. We consider the performance bottleneck on the $SM$ side, so we abandon the signature scheme. However, for $GW$ with higher computing power, we can choose appropriate signatures such as RSA, Schnorr, and Elgamal. Here we introduce the HMAC proposed by Turner, which is used to defend against key prefix and suffix attacks on MAC. Note that we believe that HMAC can achieve our required goals, and other methods can also be selected, such as NMAC and GMAC.

HMAC consists of an internal hash and an external hash. 0 is used to fill the left side of the symmetric key. The length of $K^+$ is $b$, where $b$ is the size of the block of the hash function. $HMAC$ structure can be expressed as:

$$HMAC_{key}(x) = h[(K^+ \oplus opad)||h[(K^+ \oplus ipad)||x]], \tag{5}$$

Where $x$ is the message, $h$ is the hash function, $ipad$, and $opad$ are internal padding and external padding, respectively, which are expressed as:

$$\begin{aligned} ipad &= 00110110, 00110110, ..., 00110110 \\ opad &= 01011100, 01011100, ..., 01011100, \end{aligned} \tag{6}$$

Note that a very long message $x$ is hashed only once in the internal hash function. The external hash consists only of the populated key and the internal hash. As a result, very low computation overhead is induced. The provable security of HMAC is described in section 6.

### C. Adversary Model

Based on the Dolev-Yao (DY) model, we define an adversary "$A$" whose goal is to distinguish and discover the consumption data of individual user $U$ from the communication protocol. Under the required conditions, $A$ attempts to crack the ciphertext. The adversary model includes the following hypotheses:

a) $A$ can eavesdrop on all communication records between $SM$ to $GW$ and $GW$ to $CS$. $A$ hopes to find the consumption data of $U$ from this message.

b) $A$ is a legitimate user of a network and can initiate any conversation with other entities. $A$ may try to tamper with $U$'s data through a man-in-the-middle attack or disguise $U$'s identity and send the wrong consumption data.

c) $A$ can obtain more consumption data by compromising $GW$ and $CS$, including the aggregated data plaintext after $CS$ decryption. $A$ hopes to compare the ciphertext of single and aggregated data to crack the encryption algorithm.

Based on the above description of the adversary's attack, we proceed with the following security definition.

**Definition 1: Privacy preservation.** The advantage that adversary $A$ can break the semantic security of masking is negligible.

First, to achieve pseudo-randomness in $k_i$, we can introduce a PRF $f(x)$ with indistinguishability to achieve masking. Each encryption key is uniformly selected from $\{0,1\}^\lambda$, and it is expressed as follows.

$$c_i = m_i + f(k_i) \mod M, \tag{7}$$

Based on the above equality, we can reduce semantic security to the indistinguishable property of PRFS. We then define a semantic security model of the following form (referred to as $game_1$).

The privacy preservation game 1 definition:
$d \leftarrow \{0,1\}$
$b \leftarrow \{0,1\}$
$\{ek_i\} \leftarrow \textit{Setup}(1^\lambda)$
$\{c_j = (f_{ek_{i_j}}(r_j) + m_j)\} \leftarrow A_{query}(\{i_j, m_j, r_j\})$
$c_d = M_d + f(ek_i)(w) + t_b \leftarrow D_{challenge}(M_0, M_1, w)$
$(0,1) \leftarrow A_{guess}(d')$

[2] The value of $t$ is not unique and needs to be judged according to the packet loss rate of data in the power grid, which will not be analyzed more in this paper.

Where $d$ and $b$ are both random values taken from the set $\{0,1\}$.The adversary $A$ is allowed to query $n-1$ keys $k_i$ in the setup phase. And the adversary chooses the plaintext $m_j$ and nonce value $r_j$ to request the challenger to query the ciphertext $c_j$. The challenger randomly selects one of the two messages $m_0$ and $m_1$ to encrypt and appends a blinding factor. When $b=0$, $t_b = f_{ek_n}(w)$, and when $b=1$, $t_b$ is randomly selected from $\{0,1\}^{\lambda}$.

Finally, We define the advantage of the adversary executing $D$ and $A$ algorithms to solve PRF indistinguishability and game 1 as $Adv_D^{PRF} = |Pr_D^{PRF}[Success] - 1/2|$ and $Adv_A^{game1} = |Pr_A^{game1}[Success] - 1/2|$, respectively.

**Definition 2: Authentication and Confidentiality.** The advantage of the adversary achieving a collision against the HMAC is negligible.

The definitions of collision resistance for hash and collision resistance for HMAC will be presented separately.

(1) Collision Resistance for Hash. The security of HMAC is based on the hash function $h$. For the collision-stable hash function $h(x)$, the security requirement is that adversary $A$ can find that two different messages have the same hash digest, and the advantage function of adversary $A$ is:

$$Adv_h(A) = Pr[(x, x^{'}) \leftarrow A : x \neq x^{'} and h(x) = h(x^{'})], \quad (8)$$

We use a symbol pair $(\epsilon, t)$ to denote the running time of executing the collision attack on $h$ by adversary $A$ is at most $t$ and $Adv_h(A) < \epsilon$.

We define the confidentiality of HMAC such that there exists a collision-resistant $h(x)$ against $(\epsilon, t_1)$ for which adversary $A$ has negligible attack advantage. The advantage function of the adversary and the game are defined as follows.

$$Adv_{A,S}(conf) = Pr[Game_2] = \epsilon_1, \quad (9)$$

The privacy preservation game 2 definition:
$(h, ipad, opad, key) \leftarrow \textit{KeyGen}(1^{\lambda})$
$\{c_0 = HMAC_{key}(m)\} \leftarrow A_{query}(m)$
$(c_1, c_2) \leftarrow A_{attack}(HMAC_{key_1}(x), HMAC_{key_2}(x^{'}))$
$(0,1) \leftarrow Verify(c_1, c_2)$

In game 2, adversary $A$ can ask $q$ times for the hash of any data $m$, accumulating a message block of length $\sigma$. Finally, the adversary chooses different keys and messages $x, x^{'}$ to collide with HMAC to obtain $c_1$ and $c_2$. If $c_1 = c_2$, output 1, and the adversary wins the game. Otherwise, 0 is output.

(2) Collision Resistance for HMAC. We define the authentication of HMAC such that there exists an $(\epsilon, t_1)$ collision-resistant hash such that the advantage of an adversary trying to forge the correct HMAC is negligible. Where $t_1 = t + t^{'}$, $t^{'}$ is the time for the adversary to make q queries and compute collisions. The advantage function of the adversary and the game are defined as follows.

$$Adv_{A,S}(auth) = Pr[Game_3] = \epsilon_2, \quad (10)$$

The privacy preservation game 3 definition:
$(h, ipad, opad, key) \leftarrow \textit{KeyGen}(1^{\lambda})$
$\{c_0 = HMAC_{key}(m)\} \leftarrow A_{query}(m)$
$(c_1 = HMAC_{key^{'}}(x)) \leftarrow A_{attack}(x)$
$(0,1) \leftarrow Verify(c_1, x, key)$

Similar to game 2, adversary $A$ makes $q$ queries to make forgery attempts, assuming that the length of the cumulative group of $q$ queries is $\sigma$, and the length of the adversary's forgery attempts is at most $c$.

### D. Design Goals

Based on the analysis of related works, we conclude that a practical privacy protection scheme for smart grids should meet the following objectives.

- **Lightweight**: To be adaptable to current smart meter performance, a low-overhead data encryption approach must be used.
- **Homomorphism**: To guarantee the success of encrypted data aggregation, a homomorphic encryption scheme must be employed.
- **High Fault Tolerance**: The proposed scheme must be able to tolerate a certain degree of meter failures and network delays.
- **Privacy**: The proposed scheme must ensure that data is confidential, thereby preventing adversaries from violating privacy.
- **Integrity and Authentication**: To ensure the authenticity and integrity of data, the proposed scheme should have an authentication or signature algorithm.
- **Flexibility**: To enable better data analysis, aggregate values should be more fine-grained and offer competent availability.
- **Insider Attack Resiliency**: The proposed scheme should be secure against attacks from system insiders.
- **Forward Security**: The encryption scheme should prevent all data from being decrypted due to excessive encryption times or key leakage.

For the above requirements, we propose a 3PFT scheme to achieve the first four objectives. We then apply the scheme in the aggregation environment, propose a DA-3PFT protocol to achieve the fifth, sixth, and seventh objectives, and finally design that the key update method can be deployed in DA-3PFT to achieve the last objective.

## IV. Data Aggregation Protocol For 3PFT

To achieve flexibility based on 3PFT, we propose a flexible data aggregation protocol (DA-3PFT) in this section. In this section, the overview of our protocol is first presented, followed by our proposed protocol: DA-3PFT.

### A. Overview

Our DA-3PFT is designed to build a data aggregation scheme that supports fine-grained data analysis for 3PFT. The protocol consists of five stages: 1) system initialization, 2) Report uploading, 3) Report aggregation, 4) Report extraction

TABLE II: A summary of notations

| Notation | Description |
|---|---|
| $KS$ | The key server |
| $SM$ | The smart meter |
| $GW$ | The gateway |
| $CS$ | The cloud server |
| $A$ | The adversary |
| $B$ | The challenger |
| $m_i$ | Consumption data collected by $SM$ |
| $k$ | Decryption key of $CS$ |
| $k_i$ | Masking key for $SM$ |
| $\lambda_i$ | The Lagrange coefficient |
| $k^+$ | Padded key. |
| $ipad$ | Internal padding |
| $opad$ | External padding |
| $ID_{SM}$ | The ID of $SM$ |
| $ID_{agg}$ | The ID of $GW$ |
| $(e, d)$ | A pair of RSA public and private keys |
| $m'$ | The masking consumption data |
| $m_{var}$ | The masking variance data |
| $m_{sum}$ | The aggregated data |
| $b$ | Number of data sent |
| $flag$ | Outlier marker |
| $e, d$ | A pair of RSA keys |
| $a_1, a_2 ... a_n$ | Coefficient of separating data |
| $\varepsilon$ | The probability of an adversary breaking the game |
| $L$ | Maximum length of HMAC |
| $K$ | The key used by HMAC |

and 5) load forecasting. The workflow of DA-3PFT is shown in Figure 2. During the system initialization phase, $CS$ builds the system, and KS allows $SM$ and $GW$ to register with the system. In the report upload phase, $SM$ sends data and square values to $GW$ using fault-tolerant masking methods. $GW$ can choose full or partial aggregation in the data aggregation phase according to the requirements. In the report extraction stage, $CS$ decrypts aggregate data to extract the corresponding mean and variance. In the forecasting stage, $CS$ makes long-term and short-term forecasts of future electricity consumption according to statistics. The description of some notations is given in Table 3 for ease of understanding.

### B. System Initialization

At the system initialization stage, the $CS$ first creates the system, and then the $KS$ prepares to initialize keys and register for $SM$ and $GW$.

1) **System setup**: according to a system parameter $\lambda$, $KS$ first chooses a modulo value $M$, a master key $K$ and a one-way hash function $H : \{0,1\}^* \to Z_p^*$, and sends $k$ to $CS$ (we assume $CS$ has the trust of $KS$). Then, $KS$ performs the secret sharing scheme to calculate the share $k_i$ according to the formula $k_i = f(i) = k + a_1 \cdot i^1 + a_2 \cdot i^2 + ... + a_n \cdot i^n$. It is worth mentioning that there may be too many multiplication operations in calculating functions $f(i)$. Use the Horner rule to convert the function to: $f(i) = k + i(a_1 + i(a_2 + i(...i(a_{w-1} + a_w i))))$. The time complexity can be reduced from $O(w^2)$ to $O(w)$. Finally, $KS$ selects a pair of large prime numbers $p$ and $q$, and calculates $n = pq$ and $\phi(n) = (p-1)(q-1)$.

2) **Enrollment**: the registration of $GW$ and $SM$ will be described. The registration process is performed in a secure channel, only opened during initialization to save overhead. Establishing a secure channel can refer to a key exchange protocol [40].

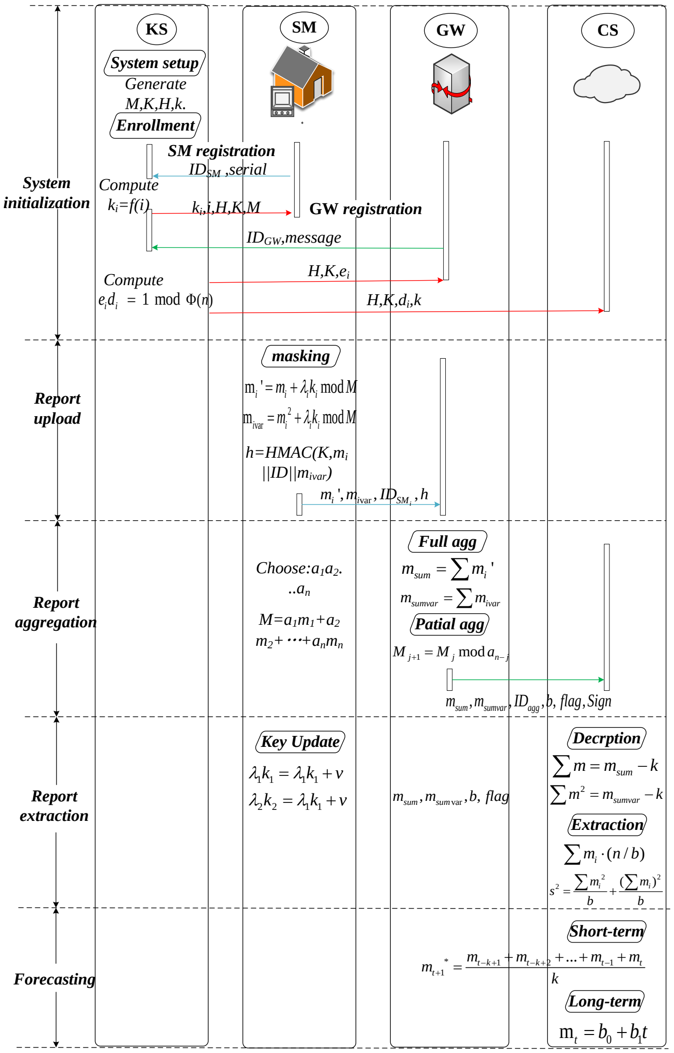

KS
SM
GW
CS
System setup
Generate M,K,H,k.
Enrollment
SM registration
$ID_{SM}$ ,serial
System initialization
Compute $k_i=f(i)$
$k_i$,i,H,K,M
GW registration
$ID_{GW}$,message
Compute $e_i d_i$ = 1 mod Φ(n)
H,K,$e_i$
H,K,$d_i$,k
Report upload
masking
$m_i' = m_i + \lambda_i k_i \bmod M$
$m_{ivar} = m_i^2 + \lambda_i k_i \bmod M$
h=HMAC(K,$m_i$ ||ID||$m_{ivar}$)
$m_i', m_{ivar}, ID_{SM_i}, h$
Report aggregation
Choose:$a_1a_2$...$a_n$
$M=a_1m_1+a_2m_2+\cdots+a_nm_n$
Full agg
$m_{sum} = \sum m_i'$
$m_{sumvar} = \sum m_{ivar}$
Patial agg
$M_{j+1} = M_j \bmod a_{n-j}$
$m_{sum}, m_{sumvar}, ID_{agg}, b, flag, Sign$
Report extraction
Key Update
$\lambda_1 k_1 = \lambda_1 k_1 + v$
$\lambda_2 k_2 = \lambda_1 k_1 + v$
$m_{sum}, m_{sumvar}, b, flag$
Decrption
$\sum m = m_{sum} - k$
$\sum m^2 = m_{sumvar} - k$
Extraction
$\sum m_i \cdot (n/b)$
$s^2 = \frac{\sum m_i^2}{b} + \frac{(\sum m_i)^2}{b}$
Forecasting
Short-term
$m_{t+1}^* = \frac{m_{t-k+1} + m_{t-k+2} + ... + m_{t-1} + m_t}{k}$
Long-term
$m_t = b_0 + b_1 t$

Fig. 2: 3PFT protocol

Step 1 ($GW$ Registration): $GW$ sends its $ID$ and registration message to $KS$. Then $KS$ selects the index $e_i \in \{1, 2, ..., \phi(n) - 1\}$ that meets the following conditions:$e_i d_i = 1 \mod \phi(n)$. This pair of keys is used to authenticate each other between $GW$ and $CS$. In addition, the key $K$ required by HAMC is also sent to $GW$ and $SM$.

Step 2 ($SM$ Registration): The $SM$ initiates a registration request $\{ID_{sm}, serialnumber\}$ to the $KS$, and the $KS$ replies to the $SM$ with the corresponding key share and supplementary information $\{k_i, i, H\}$. To control the fault tolerance rate, we can choose the number of polynomial terms $w$ in secret sharing technology, where $w = 2n/3$.

### C. Report Upload

In the report upload stage, the $SM$ collects the data $m$, then encrypts the data and attaches an authentication code.

1) **Data masking**: $SM_i$ performs masking encryption $\mathrm{m}_i' = m_i + \lambda_i k_i \mod M$. Where $\lambda_i k_i$ is Lagrange interpolation, $M$ is a sufficiently safe modulus(such as a prime number of

1024 bit), and $m_i$ is the sum of the electricity consumption of all household appliances. Because Lagrange interpolation is related to the order $i$ of users, each $\lambda_i k_i$ is calculated strictly according to their order. In addition, to represent the variance, $SM_i$ additionally calculates $\mathrm{m}_{i\mathrm{var}} = {m_i}^2 + \lambda_i k_i \mod M$. ${m_i}^2$ will be decrypted by $CS$, and the variance will be calculated in the extraction stage.

2) **Authentication**: $SM_i$ calculates the HMAC value and sends it to the corresponding gateway $GW$. Expressed as: $\{m_i^{'}, m_{i\mathrm{var}}, ID_{SM_i}, h[(K^+ \oplus opad)||h[(K^+ \oplus ipad)||m_{\mathrm{i}}^{'}||ID_{SM_i}||m_{i\mathrm{var}}\ ]]\}$, where $ID_{SM_i}$ denotes $SM's$ identity information and $K$ is the key shared by $SM$ and $GW$ in the initialization stage.

### D. Report Aggregation

At this stage, the $GW$ first verifies the integrity of the reports collected from the $SM$, calculates whether $HMAC$ are consistent with $HMAC^* = h[(K^+ \oplus opad)||h[(K^+ \oplus ipad)||m_{\mathrm{i}}^{'}{}^*||ID_{SM_i}{}^*||m_{i\mathrm{var}}{}^*\ ]]$. If not, the data is rejected. If check passes, $GW$ performs aggregation, which can be divided into full and partial aggregation according to the aggregation type.

(1) **Full aggregation**: $GW$ confirms the amount of data collected $b$ and aggregates all data to calculate $m_{sum} = \sum m_i{}'$, $m_{sumvar} = \sum m_{ivar}$, where $b$ is the actual number of $SM$ uploaded. Then, $GW$ signs the message with key $d$. $Sign = (m_{sum}||m_{sumvar}||b||flag||ID_{agg})^d$. RSA signature is used instead of HMAC because $GW$ has enough performance to execute, and the advantage of signature is non-repudiation. In addition, the short public index $e$ can be used for fast encryption. For example, $e = 2^{16} + 1$. Finally, $GW$ sends message $\{m_{sum}, m_{sumvar}, ID_{agg}, b, flag, Sign\}$ to $CS$. If $flag = 1$, the data is an outlier.

Here we discuss the judgment of outliers. We hope that $GW$ can filter the maximum and minimum values. Obviously, there is difficult to judge the extreme value of the masking. Fortunately, $GW$ can make a fuzzy judgment on the user's data. For example, if an $SM$ whose historical ciphertext is $\mathrm{m}_i{}' = m_i + \lambda_i k_i \mod M$, two situations will occur if an abnormally large value is generated. a) The large outliers cause the ciphertext to be much larger than the historical value. b) Outliers exceeding the $M$ value are remaindered at a small value. In any case, $GW$ can quickly find and filter the value. In addition, if $CS$ wants to get the data size order of $SM$, the masking scheme in this paper is difficult to achieve. It can be achieved by omitting the modulo operation, which we do not recommend because it will reduce the security of masking, and the adversary will learn more information.

(2) **Partial aggregation**: According to the existing literature, the $SM$ can measure the consumption of each piece of furniture [41]. Such as refrigerator (rf), dishwasher (dw), air conditioner (ac), and stove (st). We can describe the data collected by a $SM_i$ as $m_i = m_{rf} + m_{dw} + m_{ac} + m_{st}$. It can be understood that when the meter uploads data, it first aggregates the data of electrical appliances. It is necessary to distinguish the data to meet the fine-grained aggregation. The form is $m = a_1 \cdot m_{rf} + a_2 \cdot m_{dw} + a_3 \cdot m_{ac} + a_4 \cdot m_{st} \mod N$, Where $a_1, a_2, a_3$ and $a_4$ satisfy $a_1 > a_2 \cdot m_{dw}$, $a_2 > a_3 \cdot m_{dw}$, $a_3 > a_4 \cdot m_{dw}$, and $N$ is a modulus much larger than $a_i$. $CS$ can separate $m_j$ in the following ways:

$$\begin{aligned} & m \mod a_4 \\ & = a_1 \cdot m_{rf} + a_2 \cdot m_{dw} + a_3 \cdot m_{ac} + a_4 \cdot m_{st} \mod a_4 \\ & = a_1 \cdot m_{rf} + a_2 \cdot m_{dw} + a_3 \cdot m_{ac}. \end{aligned} \tag{11}$$

$CS$ repeats the above steps to calculate each data $m_j$, and this method can be applied to various instances of partial aggregation. Any aggregated data is separated according to this method if such fine-grained demand exists. For example, companies want data for specific buildings, and manufacturers want to get data for specific furniture. The algorithm for separating data is represented as follows:

**Algorithm 1** Data separation algorithm.

```
Input: Parameter $a_1, a_2, .., a_{n-1}, a_n$, aggregate data $M$;
Output: $m_1, m_2, ..., m_{n-1}, m_n$;
function SEPARATION($a[1:n], M, n$)
    $j = 0, M_j = M$;
    $M_{j+1} = M_j \mod a_{n-j}$;
    while $j < n$ do
        $m_n = M_j - M_{j+1}$
        $m[n] = m_n$
        $n = n - 1$
        $j = j + 1$
        $M_{j+1} = M_j \mod a_{n-j}$
return $m[1:n]$;
```

### E. Report Extraction

In the report extraction stage, $CS$ first validates and decrypts the data and then extracts aggregated data that can be analyzed.

(1) **Decryption**: The $CS$ verifies the RSA signature sent from the $GW$. Specifically, the verification equation $(m_{sum}||m_{sumvar}||b||flag||ID_{agg})^{ed} \mod n = (m^*_{sum}||m^*_{sumvar}||b^*||flag^*||ID^*_{agg})$ is established. If not, refuse. After that, $CS$ decrypts the aggregated data using the key $k$, $\sum m = m_{sum} - k$, $\sum m^2 = m_{sumvar} - k$.

(2) **Extraction**: Before analysis, $CS$ needs to perform the following operations: a) remove the outliers, b) calculate the variance $s^2 = \frac{\sum {m_i}^2}{b} + \frac{(\sum m_i)^2}{b}$, and c) optimize the aggregate value $\sum m_i \cdot (n/b)$. Specific data predictions will be analyzed next.

### F. Electric Load Forecasting

Predicting future electricity consumption series is a very important research direction in the scheduling of smart grids, which is not considered in this paper. In this section, we first analyze which situations may affect data analysis in data aggregation. Then we try to give some simple prediction methods for the reader's reference.

**Fine-grained**. Here we consider the reasons for the protocol transmission auxiliary values 1) total b, 2) square value $m^2$, and 3) outlier flag.

- **The error of aggregate data**. Owing to the complexity and uncertainty of the smart grid, the amount of data aggregated each time is different. If broken $SM$, fewer upload data do not mean less total electricity consumption. It is unscientific to judge power consumption according to the amount of aggregated data. Due to these problems, the gateway should record the amount of data collected when aggregating data.
- **Inappropriate aggregation range**. The difference in aggregation area will also affect the prediction results. For example, if the data of household consumption and shopping mall consumption data are aggregated, such prediction results are challenging to accept. The prediction result can only be the average of the two, and the different buildings should be aggregated separately. Before data aggregation, knowing the variance between single data can quickly determine whether the data is suitable for aggregation.
- **Handling of outliers**. The method based on time series is linear prediction. Abnormal data may bring a significant impact on the prediction results. Therefore, abnormal data should not be aggregated, and the gateway should filter out these data before aggregation. The encryption scheme must consider the comparison without destroying the plaintext.

Accurate load forecasting can reasonably arrange the start and stop of power generation and effectively reduce the cost of power generation. At this stage, the prediction method is divided into short-term and long-term predictions. Then, possible errors in prediction are also discussed.

**Short-term prediction**. Short-term forecasting means predicting the future of the power load in the next few hours, one day, or several days. It is used in regional power generation control, short-term resource scheduling, and market settlement. The time series method is used in the smart grid to predict future consumption. This method mainly focuses on the continuity of historical data. The method based on time series is suited well for this linear correlation prediction [42]. For example, the average value of previous data is used as the prediction result of future data. It is worth mentioning that recent data are more important than long-term data for linear correlation prediction. The prediction method is expressed as follows:

$$m_{t+1}{}^* = \frac{m_{t-k+1} + m_{t-k+2} + ... + m_{t-1} + m_t}{k}. \tag{12}$$

The data of day $t+1$ is analyzed using the data of the previous $t$ days, and $k$ is the time interval of the selected days. We choose the $k$ value that satisfies the minimum mean square for error (MSE). In addition, the machine learning method can predict nonlinear data within one day.

**Long-term prediction**. Forecasting the consumption data of the next quarter or year is used for the long-term layout of the smart grid and calculating the annual report. Unlike short-term forecasting, long-term forecasting is more concerned with the maximum and continuous load. To map long-term changes, we must consider seasonal and trend changes. For this case, we give a method to separate seasons and trends [42]. The separation prediction method is divided into three steps: First, we need to determine the seasonal coefficient to separate the seasonal components. The seasonal proportion is solved on the normalized data (by dividing by the centralized moving average). Then, it is divided by the corresponding seasonal coefficient to eliminate the influence of the season. Simple linear regression and trend analysis of consumption data. For consumption data $m$ at time $t$, the regression variance satisfies the following form:

$$\mathrm{m}_t = b_0 + b_1 t, \tag{13}$$

where $b_1 = \frac{n\sum tm - \sum t \sum m}{n\sum t^2 - (\sum t)^2}$, $b_0 = \overline{m} - b_1\bar{t}$. Finally, the trend prediction data is multiplied by the seasonal coefficient to obtain the final prediction data.

**Autoregressive Integrated Moving Average (ARIMA) model**.ARIMA model is the most common model for stationary non-white noise series data prediction. This means that the mean and variance should not change over time, and a log transformation or difference can be used to smooth the series. ARIMA includes three components: AR (autoregressive term), I (difference term), and MA (moving average term). The AR term refers to the past value used to predict the next value. Defined by the parameter $p$, where the $p$ value is derived from the Partial Autocorrelation Function (PACF) graph. The MA term defines the number of past prediction errors when predicting future values. Defined by the parameter $q$, where the Autocorrelation Function (ACF) graph is used to identify the correct $q$. The difference order specifies the number of times a sequence performs a difference operation, with the aim of keeping the data stationary. The model is divided into four steps: (1) series stationarity test and the d-order difference is performed on non-stationary time series. Determine the $p$-values and $q$-values from the PACF and ACF. (3) Fit the ARIMA model $(p, d, q)$. (4) Predict future values.

## V. A Key Update Method Based on Negotiation

To enhance the forward security of DA-3PFT, we propose a negotiation-based key update method that can be utilized in the initialization stage of the protocol. This section is divided into two parts: first, we discuss the potential attacks that can affect DA-3PFT, and then we present our key update method to counter those attacks.

### *A. Attacks on DA-3PFT*

We consider three attacks that can be launched on DA-3PFT: key leakage attack, known plaintext attack, and ciphertext subtraction attack.

*1) Key Leakage Attack:* If the user's key is not properly stored and gets leaked, the attacker can decrypt all the user's past consumption data using the compromised key. Unfortunately, it is difficult for the user to request an update of the key at this point due to secret sharing restrictions.

*2) Known Plaintext Attack:* The masking approach utilized in DA-3PFT lacks the addition of random variables, which makes the encryption result predictable. An attacker can exploit this vulnerability by intercepting the ciphertext and looking for a plaintext-ciphertext pair ($\mathrm{m}_i{}' - m_i$) to recover the user's key. This attack can be initiated by recording meter readings and checking the ciphertext in the real world.

*3) Ciphertext Subtraction Attack:* In 3PFT, the same key $\lambda_i k_i$ is used, making it easier for an attacker to get information about the plaintext ($m_i{}^2 - m_i = \mathrm{m}_{i\mathrm{var}} - \mathrm{m}_i{}'$). This information can be used to analyze the consumption stability of users.

### B. Key Update Method

To counter the above attacks, we propose a key update method that employs negotiation-based key sharing. While the introduction suggested that a user could request multiple sets of keys from $KS$ periodically to improve masking security, this approach is not practical due to communication and storage overhead. To address this issue, we have designed a scheme that requires each user to select another user to execute the key update protocol.

Before presenting the protocol, we classify aggregation into two categories: horizontal aggregation and vertical aggregation (see Fig. 3). For the first vertical aggregation, self-updating the key is sufficient to defend against key leakage attacks. For example, a random number $v$ can be generated by the $SM$ or the user and used to update the keys of $\lambda_1 k_1$ and $\lambda_2 k_2$ by adding and subtracting $v$, respectively. For horizontal aggregation, not all users are cooperative and willing to update their keys. To account for this, we designed a flexible user-specific approach that allows individual users to select another user and initiate the key update protocol without affecting other users. Moreover, to prevent the forgery of user identity, we have added a signature scheme to the protocol process. Although this scheme results in additional overhead, it is necessary without affecting the efficiency of data uploads. Further optimization is possible, with users interacting with terminal devices and inputting negotiation results into the meter to save computational resources for the $SM$.

The protocol consists of two steps: first, the users share their keys through negotiation, and then they generate privacy modification parameters $R_i$ and $R_j$ based on their keys. The two users then modify $\lambda k$ by consensus. The protocol is based on the assumption that $U_i$ and $U_j$ have a pair of public and private keys, respectively.

Step 1 (key negotiation): Each user negotiates using the authenticated DH protocol [43] as follows:

- User $U_i$ sends a key update request to user $U_j$. If $U_j$ accepts, it selects a random number $RA_j$ and sends it to $U_i$.
- $U_i$ selects two random numbers, $a_i$ and $RA_i$, and calculates $A_i = g^{a_i}$. Then, $U_i$ uses these values to compute the corresponding signature $sign(H(A_i)||RA_j)$ and sends $\{A_i, RA_i, sign\}$ to $U_j$. Any RSA or Elgamal signature scheme can be used; no specific requirement exists for the signature.
- After $U_j$ verifies, it selects a random number $b_j$ and calculates $B_j = g^{b_j}$. Similarly, $U_j$ sends $\{B_j, RA_j, sign(H(B_j)||RA_i)\}$ to $U_i$.

Step 2 (blinded value change): For the fairness of data modification, each user on both sides provides a modification parameter. Users $U_i$ and $U_j$ select random numbers $R_i$ and $R_j$, respectively, and use key $L_{ij}$ to encrypt and send them to each other. Here, the encryption algorithm is symmetric encryption, such as AES and SM4. After that, $U_i$ and $U_j$ update their pairs of keys to:

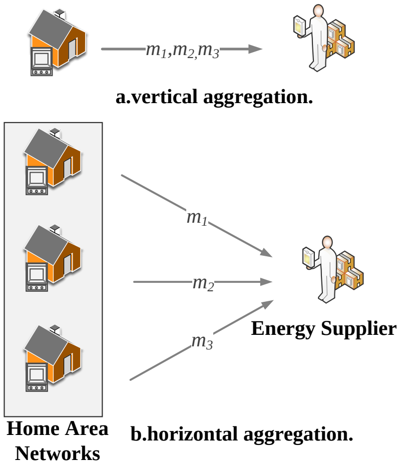


Fig. 3: Two types of aggregation

$$\begin{aligned} \lambda_i k_i{}' &= \lambda_i k_i - R_i + R_j, \\ \lambda_j k_j{}' &= \lambda_j k_j + R_i - R_j. \end{aligned} \tag{14}$$

Notably, our protocol only changes the value of $\lambda_i k_i$ and does not affect the value of the secret $k$. The modified key will revert to the state of the original key in the aggregation process. In addition, some schemes can guarantee to change $k_i$, but $k$ is unchanged [44]. Such methods are useful for some scenarios, such as distributed secret sharing, and are not considered in this paper.

## VI. Security Analysis

This section mainly analyzes the security of the proposed 3PFT, DA-3PFT, and key update methods on the aspects of fault tolerance, privacy protection, authentication, and correctness.

### A. Formal Security Proof

**Theorem 1** (Privacy preservation)**.** . *As in Definition I, if adversary A can guess the consumption data of a single user from the encrypted data, we can construct an algorithm D to break the indistinguishable property of PRF.*

*Proof.* To facilitate our proof, in $game_1$, We use $X_d$ to represent $M_d + f_{ek_i}(x)$, and the challenge ciphertext can be expressed as $c_d = X_d + t_d$. The probability that algorithm $D$ can successfully distinguish between $f_{sk_n}(w)$ and a random number is:

$$\begin{aligned}Pr_{A'}^{PRF}[Suc] &= Pr[b' = b]\\ &= \frac{1}{2}\{Pr[b' = 0|b = 0] + Pr[b' = 1|b = 1]\}\\ &= \frac{1}{4}\{Pr[A(t_0 + X_0) = 0] + Pr[A(t_0 + X_1) = 1]\\ &+ Pr[A(t_1 + X_0) = 0] + Pr[A(t_1 + X_1) = 1]\}\\ &= \frac{1}{4}\{2Pr_A^{g_1} + 1 - (Pr[A(t_1 + X_0) = 1]\\ &- Pr[A(t_1 + X_1) = 1])\}\end{aligned} \tag{15}$$

The probability that the adversary $A$ succeeds in winning the game1 is:

$$Pr_A^{g_1} = \frac{1}{2}Pr[A(t_0 + X_0) = 0] + \frac{1}{2}Pr[A(t_0 + X_1) = 1] \tag{16}$$

Combining the previous relationship between advantage and probability and taking the absolute value, we get:

$$\begin{aligned}&2Adv_D^{PRF} + \frac{1}{2}|Pr[A(t_1 + X_0) = 1] - Pr[A(t_1 + X_1) = 1]|\\ &\geq Adv_A^{g_1}\end{aligned} \tag{17}$$

Since $t_0 + X_0$ and $t_0 + X_1$ are identically distributed, it follows that $Pr[A(t_1 + X_0) = 1] = Pr[A(t_1 + X_1) = 1]$. We have:

$$2Adv_D^{PRF}(\lambda) \geq Adv_D^{g_1}(\lambda) \tag{18}$$

From the above formula, we can conclude that: if $Adv_D^{g_1}$ is non-negligible, we can use the $D$ algorithm to achieve the indistinguishability of PRF with the same advantage. This is in conflict with the assumption that PRF is indistinguishable, so $Adv_D^{g_1}$ is negligible. □

In practice, an adversary may live in a residential area and eavesdrop on reports sent by users to the gateway, thereby being able to perform a chosen ciphertext attack. For a data $m_i$, the adversary must attempt to crack the ciphertext $CT$ within 15 minutes. Whether it is brute-force $m_i$ or trying to analyze the key, it is semantically secure that masking based on the above model is secure for chosen ciphertext attacks without knowing the key. Therefore, even if an adversary eavesdrops on the $CT$, it cannot identify the user's content. Additionally, $GW$ does not decrypt user reports during the protocol but aggregates them directly. Thus, even if adversary $A$ can hack into $GW's$ database, it can only obtain ciphertext data. In addition, if adversary $A$ breaks into the $CS's$ database, it can only get aggregate data, not individual user data.

**Theorem 2** (Authentication and integrity protection)**.** . *In DA-3PFT, if adversary $A$ tampers with data or sends data by forging an identity, we can construct an algorithm $D$ to break the collision resistance of the hash function.*

*Proof.* The proof begins by defining a $(\varepsilon, t, q, L)$-security Message Authentication Code (MAC) as presented in [45]. If the adversary does not have the key $k$, it should attack within a limited time $t$ and select a query with at most $q$ messages, where each query has a message length of $L$. The adversary cannot break the scheme except with probability better than $\varepsilon$. In our protocol, we use the SHA-512, which is a unidirectional function and satisfies $(\varepsilon, t, q, L)$-security.

The process of this security reduction is obvious. Assuming that the adversary can achieve winning and breaking $(\varepsilon, t, q, L)$-security in Game 2, the adversary can construct the following equation:

$$h[B||h[(A||x)]] = h[B^*||h[(A^*||x')]], \tag{19}$$

Algorithm $D$ can construct a set of messages $m = B||h[(A||x)]$ and $m' = B^*||h[(A^*||x')]$ satisfying the collision hash based on the above equality, which breaks the assumption that SHA-512 is secure and indistinct. Therefore, the advantage of the adversary solving the collision resistance of HMAC is negligible.

□

### B. Informal Security Analysis

(1) **Fault Tolerance.** Some meters may cause errors when transmitting consumption data due to physical damage or adversary intrusion. A good protocol should be error tolerant and aggregate correctly. In our protocol, $GW$ only collects $t$ data and discards $n-t$ data without affecting the aggregation. When the rest of the $n-t$ data is lost or compromised, $CS$ can still decrypt the aggregated data according to the following formula.

$$\begin{aligned}m_{agg} &= m_1 + m_2 + ... + m_t + \lambda_1 k_1 + \lambda_2 k_2 + ... + \lambda_i k_i\\ &= \sum_1^t m_i + k.\end{aligned} \tag{20}$$

It is clear that based on the properties of the Lagrange interpolation theorem, only enough $t$ data can be aggregated into a key $k$. In other words, DA-3PFT can guarantee the correct aggregation results on the fault-tolerant mechanism. At the same time, additional computation and communication costs are reduced.

(2) **Insider Attack Resilient.**An internal adversary can compromise $GW$ and $CS$'s and obtain their keys and decrypt consuming data. Unfortunately, based on the mechanism of aggregation, $GW$ aggregates the data in the ciphertext case, and the key of $CS$ can only decrypt the aggregated data. The adversary does not gain many advantages over $CS$ and $GW$, so it is difficult for the adversary to break the consumption data of a single uncompromised $SM$.

(3) **Correctness of Key Update.** If the negotiation process is valid, the key update method is considered to have been executed correctly. In the key update method, the security of key negotiation is based on the security of DH protocol, which can be defined as the decisional Diffie-Hellman Problem (DDH) [46]. In addition, the signature also prevents the possible man-in-the-middle attack of DH protocol. Similarly, if the DDH problem is difficult, two users can safely exchange keys during negotiation. Similarly, if solving the DDH problem is difficult,

two users can safely exchange keys during negotiation. And we can prove equation $\sum k_{change} = \sum k_i$ holds.

$$\begin{aligned}&\lambda_i {k_i}' + \lambda_j {k_j}' \\ &= \lambda_i k_i - R_i + R_j + \lambda_j k_j + R_i - R_j \mod M \qquad (21) \\ &= \lambda_i k_i + \lambda_j k_j \mod M.\end{aligned}$$

Finally, the proof argues that if the key update is performed incorrectly, such that $\lambda_i {k_i}'$ and $\lambda_j {k_j}'$ are random numbers, then the probability that they can be verified correctly is $1/M$, where $M$ is the modulus. As this probability can be ignored, incorrect keys will be discarded during the aggregation. Therefore, it can be concluded that the key update function is executed correctly if the negotiation process is valid.

## VII. Performance Analysis

In this section, we first conduct a theoretical analysis to compare the computation cost and communication cost of DA-3PFT with other protocols. After that, we present a simulation platform to simulate the actual computation cost and support the above analysis results.

We compare the performance of the proposed scheme with five benchmark schemes: LPDA [12], Zuo [47], BAMDD [48], FTMA [17], and LCEDA [26]. These works used different encryption techniques such as masking, ElGamal, and Pailliar. The notations used in Table III and Table IV are defined as follows.

TABLE III: Time of operation in the experiment

| Notation | Time of operation |
|---|---|
| $T_{add}$ | Modular addition |
| $T_{xor}$ | Modular XOR |
| $T_{mul}$ | Modular multiplication in $Z_{n^2}$ |
| $T_h$ | SHA-256 hash function |
| $T_e$ | Modular exponentiation in a cyclic group[1] |
| $T_p$ | Pairing |
| $T_{pm}$ | Point multiplication in ECC[2] |
| $T_{epa}$ | Pailiar encryption |
| $T_{aes}$ | Advanced encryption standard |

[1] e.g. $G_T$ in [49]

[2] e.g. typeA in JPBC

TABLE IV: Number of entities in the experiment

| Notation | Number of entities |
|---|---|
| $l$ | Household appliances of a user |
| $w$ | $SM$ accommodated in $GW$ |
| $c$ | $GW$ governed by the $CS$ |

### A. Theoretical Evaluation

The theoretical evaluation includes the protocol's computational cost and communication cost.

(1) **Computational cost**. Table V shows the entities' computational costs of different privacy-preserving protocols. For DA-3PFT, in the report upload stage, the $SM$ calculates the consumption data and its square value and encrypts it. The time to calculate $\mathrm{m}_i{}' = m_i + \lambda_i k_i \bmod M$, $\mathrm{m}_{i\mathrm{var}} = m_i{}^2 + \lambda_i k_i \bmod M$ is $2T_{add} + 3T_{mul}$. Addition and multiplication are simple operations and will not consume more computing time. The time to calculates $HMAC = h[(K^+ \oplus opad)||h[(K^+ \oplus ipad)||m_{\mathrm{i}}^{'}||ID_{SM_i}||m_{i\mathrm{var}}]]$ to achieve authentication is $2T_{xor} + 2T_h$. XOR and hash operations are also simple operations that take less time. In the report aggregation stage, the time for the $GW$ to verify the HMAC $HMAC^* = h[(K^+ \oplus opad)||h[(K^+ \oplus ipad)||m_{\mathrm{i}}^{'*}||ID_{SM_i}{}^*||m_{i\mathrm{var}}{}^*]]$ is $2wT_{xor} + 2wT_h$ and the time for calculating the signature of the sent message $(m_{sum}||m_{sumvar}||b||flag||ID_{agg})\}^d$ is $wT_e$. In contrast, modular exponentiation takes some time. However, this is effortless for the performance of the $GW$. RSA authentication is also a lightweight two-way authentication method. In the report extraction stage, the calculation time for $CS$ to verify the RSA signature $(m_{sum}||m_{sumvar}||b||flag||ID_{agg})^{ed} \bmod n$ is $cT_e$, and the calculation time for decrypting the aggregate value $\sum m = m_{sum} - k$, $\sum m^2 = m_{sumvar} - k$ is $2cT_{add}$. We have also performed statistics on computational operations for other protocols.

Based on the information presented in Table V, it is evident that the computation cost of $SM$ in the LPDA scheme is the lowest, $GW$ in the FTMA scheme has the lowest computation cost, and $CS$ in the LCEDA scheme exhibits the lowest computation cost. And our protocol achieves a balance across all metrics, ensuring that the operations performed by each entity remain lightweight. This advantageous outcome can be attributed to the absence of high-complexity operations such as $T_p$ and $T_{epa}$ in our protocol.

To further explore and compare the computational cost of each protocol, we have conducted an experimental analysis where we simulate and measure the computational requirements. This approach enables us to visually depict and scrutinize the discrepancies in computational cost among the different protocols under consideration.

(2) **Communication cost**. We set a reasonable size for each symbol to compare the communication cost. The communication cost includes two parts, the communication cost from the $SM$ to the $GW$ and the communication cost from the $GW$ to the $CS$.

Table VI shows the communication cost comparison of different schemes. In the report upload, $SM$ sends $\{m_i^{'}, m_{i\mathrm{var}}, ID_{SM_i}, h\}$ to $GW$, and the communication cost is $512 + 512 + 32 + 256 = 1312b$. In the aggregation stage, $GW$ sends message $\{m_{sum}, m_{sumvar}, ID_{agg}, b, flag, Sign\}$ to $CS$, and the communication cost is $512+512+32+16+1+1024 = 2097b$. In scheme [12], $SM$ sends $\{tid_i, X_i, H_i, t_i\}$ to $GW$ in step AG1, and the communication cost is $32+512+256+64 = 864b$. $GW$ sends message $\{tid_j, X_j, H_j, t_j, E_k\}$ to $CS$ in step AG2, and the communication cost is $32 + 512 + 256 + 64 + 256 = 1120b$. In scheme [47], $SM$ sends $\{ID_i, C_i^a, C_i^b, T_i, \sigma_i\}$ to $GW$ in encryption of user data stage, and the communication cost is $32 + 512 + 512 + 64 + 1024 = 864b$. $GW$ sends message $\{ID_{GW}, C^a, C^b, T_{GW}, \sigma_{GW}\}$ to $CS$ in the data aggregation stage, and the communication cost is $32 + 512 + 512 + 64 + 1024 = 2144b$. In scheme [48], $SM$ sends $\{C_i, g_2^{sk_i}, g_2^{sk_{U_i}}, T, \sigma_i\}$ to $GW$ in reports generation, and the communication cost is $2048+512+512+64+160 = 3296b$. $GW$ sends message $\{tid_j, X_j, H_j, t_j, E_k\}$ to $CS$ in data aggregation, and the communication cost is

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3303010

TABLE V: Computational costs of different privacy-preserving protocols

| Protocol | SM | GW | CS |
|---|---|---|---|
| **DA-3PFT** | $2T_{xor} + 2T_h + 2T_{add} + 3T_{mul}$ | $w(2T_{xor} + 2T_h + T_e)$ | $c(T_e + 2T_{add})$ |
| LPDA [12] | $T_{add} + T_h$ | $w(T_{xor} + 2T_{add} + T_h + T_{aes})$ | $cw(T_{add} + T_{mul})$ |
| Zuo et al.'s [47] | $5T_e + (4w - 4 + l)T_{mul} + (w + 2)T_h + 2T_p + T_{pm}$ | $(w + 1)T_p + 3wT_{mul} + T_h + T_e$ | $cw(2T_p + T_h) + c(w - 1)T_{mul}$ |
| BAMDD [48] | $(l + 1)T_{mul} + T_{epa} + T_h$ | $4T_p + wT_{mul} + T_h$ | $6T_p + cT_{mul} + T_h$ |
| FTMA [17] | $4T_e + 8T_{mul} + 4T_h + 3T_p$ | $wT_{mul}$ | $c(T_e + T_{mul}) + 2cwT_{add}$ |
| LCEDA [26] | $(w - 1)T_{poly} + 2(w - 1)T_{mul} + (w + 2)T_{add}$ | $(w - 1)T_{add}$ | $cT_{add}$ |

TABLE VI: Communication costs of different privacy-preserving protocols

| Protocol | SM to GW (b) | GW to CS (b) |
|---|---|---|
| **DA-3PFT** | $1312w$ | $2097c$ |
| LPDA [12] | $864w$ | $1120c$ |
| Zuo et al.'s [47] | $2144w$ | $2144c$ |
| BAMDD [48] | $3296w$ | $3296c$ |
| FTMA [17] | $2048w$ | $256c$ |
| LCEDA [26] | $1312w$ | $1312c$ |

$2014 + 512 + 512 + 64 + 160 = 3296b$. In scheme [17], $SMs$ sends $\{pk_i, r_i P_0\}$ to each other and computes ciphertext $c_i$ to $GW$, and the communication cost is $1024 + 256 + 256 = 2048$. $GW$ sends ciphertext aggregation $C$ to $CS$, and the communication cost is 256. Note that this scheme does not consider the authentication between the individual entities, so this communication cost does not include the cost of signing, which may be more expensive in practice than in theory. In scheme [26], $SM$ sends ciphertext and identity $\{c_i, ID, sign\}$ to $CS$, and the communication cost is $256{+}32{+}1024 = 1312$. $GW$ updates the polynomial and sends the aggregate ciphertext $\{C, ID, sign\}$ to $CS$, and the communication cost is $256 + 32 + 1024 = 1312$.

According to the results in Table VI, we can see that our scheme has certain advantages in the communication of the protocol. The work [12] only supports one data type and does not support implementing complex data analysis. To achieve flexibility in data analysis, the mean and variance sent in this paper cause a certain additional communication cost, which we believe is worth it. In addition, according to the design objectives, we compare the properties of multiple schemes, and the comparison results are shown in Table VII. Table VII shows that our scheme satisfies more properties than other schemes. This scheme is lightweight because it does not use high-overhead homomorphic encryption and relies on secret sharing technology to achieve high fault tolerance. In addition, forward security and insider attacks increase the protocol's robustness.

### B. Experimental Evaluation

The server configuration used in the experiment is as follows: an Intel 3.0GHz i5-8500 CPU, 16GB RAM, and Windows 10. And we use JPBC (Java Pairing-Based Cryptography Library) to implement our concerned key cryptographic operations in the protocol, in which RSA modulus and $n$ are 1024 bits, and the hash function is SHA-256.

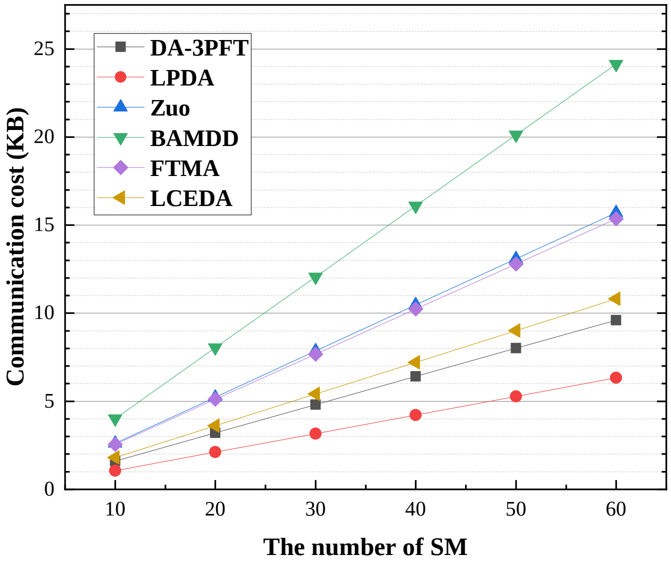


Fig. 4: Comparison of communication cost on $SM$

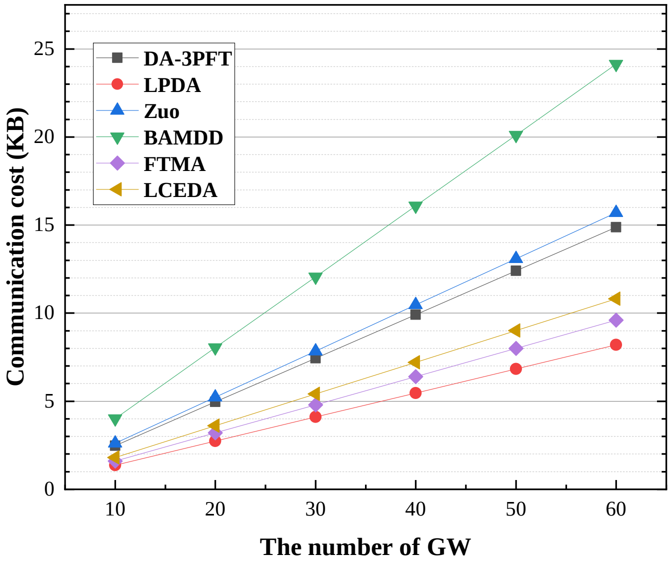


Fig. 5: Comparison of communication cost on $GW$

First, we simulate the communication cost, using different computers to simulate the sending of data between entities in the protocol and measure the communication cost. The experimental results are shown in Fig 4 and Fig 5. These two figures illustrate that the communication cost of all schemes grows linearly before a single entity reaches the performance bottleneck. In Fig 4, as the number of $SM$ increases, the communication cost of LPDA is minimal, and only encrypted data is delivered. BAMDD has the highest communication cost because the authentication process is more complex and requires the $SM$ to cooperate with each other. In contrast, our DA-3PFT scheme has a large advantage and satisfies more

TABLE VII: A comparison of existing literature

| Functions | [18] | [48] | [9] | [12] | [17] | [16] | [50] | our shceme |
|---|---|---|---|---|---|---|---|---|
| Lightweight | × | × | √ | √ | × | √ | × | √ |
| Homomorphism | √ | √ | × | √ | √ | √ | √ | √ |
| High fault tolerance | × | × | × | × | √ | × | √ | √ |
| Authentication and integrity | √ | √ | √ | √ | − | √ | √ | √ |
| Flexibility | √ | × | × | × | √ | × | √ | √ |
| Insider attack resiliency | √ | √ | √ | × | √ | √ | √ | √ |
| Forward security | − | − | √ | √ | − | − | − | √ |

Note: For different functions, √ means support; × means not supported; − means not mentioned.

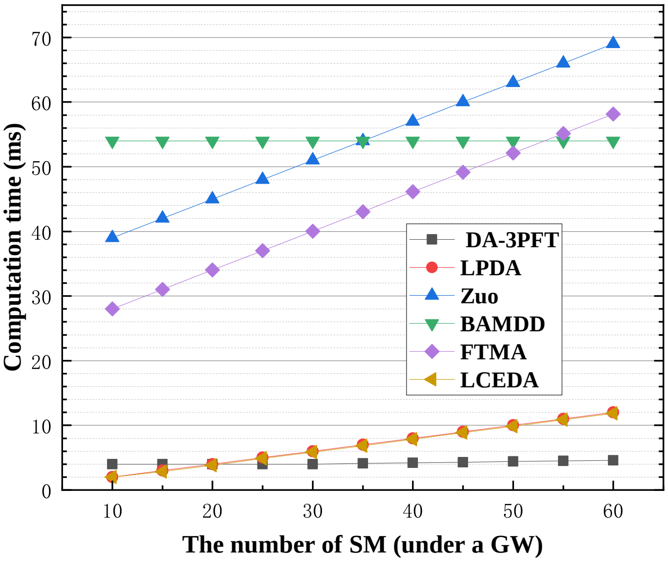


Fig. 6: Comparison of time cost on $SM$

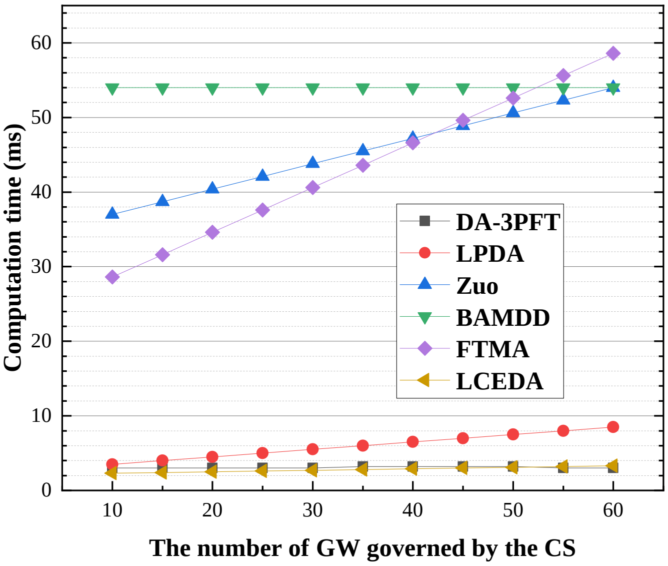


Fig. 7: Comparison of time cost on $GW$

properties among the five schemes. In Fig 5, as the number of $GW$ increases, LPDA has the smallest communication cost. LPDA does not consider the authentication scheme in the protocol, so this part of the cost is not taken into account. BAMDD has the largest communication cost due to authentication complexity and pairwise encrypted data. In contrast, the communication cost of our DA-3PFT is in the middle because the mean and variance are transmitted in the protocol, which increases the amount of transmitted data.

$SM$ and $CS$ are required to have low computational overhead compared to other entities. The reduced encryption computation cost allows for faster and more efficient transmission of data between $SM$ and $CS$. Based on the JPBC library, we simulate the overall computational cost of each scheme to compare them in detail. For Fig 6, we assume that there is a case of 1 $GW$ and 1 $CS$ and measure the change of computational cost as $SM$ increases. For Fig 7, we assume the presence of 10 $SMs$ and 1 $CS$ and measure the change of computational cost as $GW$ increases. From the overall experimental results, the computational cost of the protocol can be roughly divided into two categories. One is that the computational cost remains stable or increases slightly with the increase of entities, and the other is that the computational cost increases greatly with the increase of entities. It can be seen from the figure that DA-3PFT, LPDA, and LCEDA schemes have smaller computational costs, and the computation time is less than 10ms. The computational cost of the BAMDD scheme does not increase with the increase in the number of entities, but the initial cost of the scheme is relatively large, which is not suitable for small-scale power grid scenarios. In contrast, the DA-3PFT scheme has the advantage of low overhead and stability because the scheme in this paper does not adopt large overhead $T_p$ and $T_{pm}$ operations.

Finally, we simulate each stage of the protocol: report upload, report aggregation, and report extraction. The stage names of other protocols may be different, but they can still be divided into these three phases according to the method of our protocol. Because each protocol's initialization (registration) phase is very different, this phase is prepared first, which does not affect the data upload time. We do not take this part into account. Moreover, the entities in the initialization phase are third-party trusted entities, which can fully bear the overhead of this phase. As can be seen in Fig 8, the upload phase of the LPDA scheme has the smallest computational cost, and the LCEDA scheme has the smallest cost in the aggregation and extraction phases. Both FTMA and BAMDD schemes suffer from excessive overhead in some phases. In contrast, our DA-3PFT scheme is more stable in all stages and satisfies more

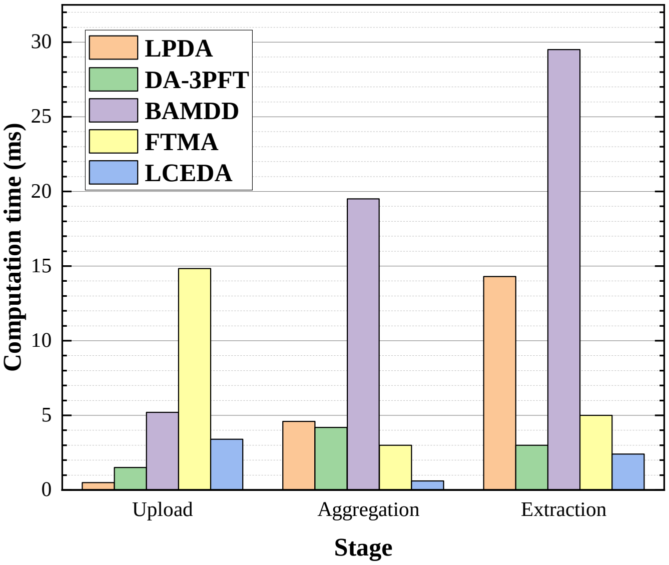


Fig. 8: Comparison of time cost in different stages

properties.

## VIII. Conclusion

This paper proposed a practical privacy-preserving scheme with fault tolerance (3PFT) in the smart grid. In 3PFT, Shamir's secret sharing technique is applied in the masking approach that requires only part of the data to recover the master key. Next, we devise a flexible data aggregation protocol (DA-3PFT) for 3PFT to support load forecasting. Besides, a negotiation-based key update method is designed to improve the forward security of the DA-3PFT. Lastly, our security and performance evaluations demonstrated the utility of DA-3PFT.

## References

[1] S. Wang, A. F. Taha, J. Wang, K. Kvaternik, and A. Hahn, "Energy crowdsourcing and peer-to-peer energy trading in blockchain-enabled smart grids," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 8, pp. 1612–1623, 2019.

[2] Y. Su, J. Li, Y. Li, and Z. Su, "Edge-enabled: A scalable and decentralized data aggregation scheme for iot," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2022, 10.1109/TII.2022.3170156.

[3] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *2010 first IEEE international conference on smart grid communications*. IEEE, 2010, pp. 238–243.

[4] M. S. Haghighi and Z. Aziminejad, "Highly anonymous mobility-tolerant location-based onion routing for vanets," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2582–2590, 2019.

[5] D. Varodayan and A. Khisti, "Smart meter privacy using a rechargeable battery: Minimizing the rate of information leakage," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 1932–1935.

[6] D. Riboni, L. Pareschi, and C. Bettini, "Js-reduce: Defending your data from sequential background knowledge attacks," *IEEE Transactions on dependable and Secure Computing*, vol. 9, no. 3, pp. 387–400, 2012.

[7] H. Shen, M. Zhang, and J. Shen, "Efficient privacy-preserving cube-data aggregation scheme for smart grids," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1369–1381, 2017.

[8] 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 of Things Journal*, vol. 6, no. 3, pp. 4755–4763, 2018.

[9] P. Parmar and B. Kadhiwala, "Secure data aggregation protocol using aes in wireless sensor network," in *Emerging Research in Computing, Information, Communication and Applications*. Springer, 2016, pp. 421–432.

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

[11] J. N. Liu, J. Weng, A. Yang, Y. Chen, and X. Lin, "Enabling efficient and privacy-preserving aggregation communication and function query for fog computing based smart grid," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 247–257, 2019.

[12] P. Gope and B. Sikdar, "Lightweight and privacy-friendly spatial data aggregation for secure power supply and demand management in smart grids," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1554–1566, 2018.

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

[14] F. Gomez Marmol, C. Sorge, R. Petrlic, O. Ugus, D. Westhoff, and G. Martínez Pérez, "Privacy-enhanced architecture for smart metering," *International journal of information security*, vol. 12, no. 2, pp. 67–82, 2013.

[15] M. Jang, K. Nam, and Y. Lee, "Analysis and application of power consumption patterns for changing the power consumption behaviors," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 25, no. 4, pp. 603–610, 2021.

[16] I. A. Kamil and S. O. Ogundoyin, "Epdas: Efficient privacy-preserving data analysis scheme for smart grid network," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 2, pp. 208–217, 2021.

[17] X. Wang, Y. Liu, and K.-K. R. Choo, "Fault-tolerant multisubset aggregation scheme for smart grid," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4065–4072, 2020.

[18] A. Mohammadali and M. S. Haghighi, "A privacy-preserving homomorphic scheme with multiple dimensions and fault tolerance for metering data aggregation in smart grid," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 5212–5220, 2021.

[19] C. Xu, R. Yin, L. Zhu, C. Zhang, C. Zhang, Y. Chen, and K. Sharif, "Privacy-preserving and fault-tolerant aggregation of time-series data with a semi-trusted authority," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12 231–12 240, 2022.

[20] L. Wu, M. Xu, S. Fu, Y. Luo, and Y. Wei, "Fpda: Fault-tolerant and privacy-enhanced data aggregation scheme in fog-assisted smart grid," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5254–5265, 2022.

[21] L. Wu, S. Fu, Y. Luo, H. Yan, H. Shi, and M. Xu, "A robust and lightweight privacy-preserving data aggregation scheme for smart grid," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–14, 2023.

[22] A. Saleem, A. Khan, S. Malik, H. Pervaiz, H. Malik, M. Alam, and A. Jindal, "Fesda: Fog-enabled secure data aggregation in smart grid iot network," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6132–6142, 2020.

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

[24] J. Song, Y. Liu, J. Shao, and C. Tang, "A dynamic membership data aggregation (dmda) protocol for smart grid," *IEEE Systems Journal*, vol. 14, no. 1, pp. 900–908, 2019.

[25] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2018.

[26] Y. Su, Y. Li, J. Li, and K. Zhang, "Lceda: Lightweight and communication-efficient data aggregation scheme for smart grid," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15 639–15 648, 2021.

[27] S. Wang, L. Huang, Y. Nie, X. Zhang, P. Wang, H. Xu, and W. Yang, "Local differential private data aggregation for discrete distribution estimation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 9, pp. 2046–2059, 2019.

[28] M. Yang, I. Tjuawinata, K. Y. Lam, J. Zhao, and L. Sun, "Secure hot path crowdsourcing with local differential privacy under fog computing architecture," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2188–2201, 2022.

[29] D. Ye, S. Shen, T. Zhu, B. Liu, and W. Zhou, "One parameter defense—defending against data inference attacks via differential privacy,"

*IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1466–1480, 2022.

[30] J. Wei, Y. Lin, X. Yao, and J. Zhang, "Differential privacy-based location protection in spatial crowdsourcing," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 45–58, 2022.

[31] B. Jayaraman and D. Evans, "Evaluating differentially private machine learning in practice," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*. USENIX, 2019, pp. 1895–1912.

[32] Y. Chen, J.-F. Martínez-Ortega, L. López, H. Yu, and Z. Yang, "A dynamic membership group-based multiple-data aggregation scheme for smart grid," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12 360–12 374, 2021.

[33] Y. Zhan, L. Zhou, B. Wang, P. Duan, and B. Zhang, "Efficient function queryable and privacy preserving data aggregation scheme in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 3430–3441, 2022.

[34] X. Yan, W. W. Y. Ng, B. Zhao, Y. Liu, Y. Gao, and X. Wang, "Fog-enabled privacy-preserving multi-task data aggregation for mobile crowdsensing," *IEEE Transactions on Dependable and Secure Computing*, 2023.

[35] P. Gupta, Y. Li, S. Mehrotra, N. Panwar, S. Sharma, and S. Almanee, "Obscure: Information-theoretically secure, oblivious, and verifiable aggregation queries on secret-shared outsourced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 843–864, 2022.

[36] J. Wang, J. Appiah-Kubi, L.-A. Lee, D. Shi, D. Zou, and C.-C. Liu, "An efficient cryptographic scheme for securing time-sensitive microgrid communications under key leakage and dishonest insiders," *IEEE Transactions on Smart Grid*, vol. 14, no. 2, pp. 1210–1222, 2023.

[37] J. Niu, X. Li, J. Gao, and Y. Han, "Blockchain-based anti-key-leakage key aggregation searchable encryption for iot," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1502–1518, 2020.

[38] Y. Chen, S. He, B. Wang, P. Duan, B. Zhang, Z. Hong, and Y. Ping, "Cryptanalysis and improvement of deeppar: Privacy-preserving and asynchronous deep learning for industrial iot," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 958–21 970, 2022.

[39] X. Zhang, C. Huang, C. Xu, Y. Zhang, J. Zhang, and H. Wang, "Key-leakage resilient encrypted data aggregation with lightweight verification in fog-assisted smart grids," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8234–8245, 2021.

[40] K. Seyhan, T. N. Nguyen, S. Akleylek, K. Cengiz, and S. H. Islam, "Bi-gisis ke: Modified key exchange protocol with reusable keys for iot security," *Journal of Information Security and Applications*, vol. 58, no. 1, pp. 1027–1088, 2021.

[41] B. Yildiz, J. I. Bilbao, J. Dore, and A. B. Sproul, "Recent advances in the analysis of residential electricity consumption and applications of smart meter data," *Applied Energy*, vol. 208, no. 1, pp. 402–427, 2017.

[42] R. Liu, X. Li, L. Han, and J. Meng, "Track infrared point targets based on projection coefficient templates and non-linear correlation combined with kalman prediction," *Infrared Physics & Technology*, vol. 57, no. 1, pp. 68–75, 2013.

[43] N. Li, "Research on diffie-hellman key exchange protocol," in *2010 2nd International Conference on Computer Engineering and Technology*, vol. 4. IEEE, 2010, pp. 4–634.

[44] R. Vassantlal, E. Alchieri, B. Ferreira, and A. Bessani, "Cobra: Dynamic proactive secret sharing for confidential bft services," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2022, pp. 1528–1528.

[45] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Annual international cryptology conference*. Springer, 1996, pp. 1–15.

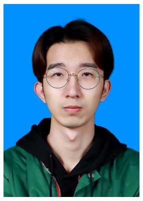
[46] H. Krawczyk, K. G. Paterson, and H. Wee, "On the security of the tls protocol: A systematic analysis," in *Annual Cryptology Conference*. Springer, 2013, pp. 429–448.

[47] X. Zuo, L. Li, H. Peng, S. Luo, and Y. Yang, "Privacy-preserving multidimensional data aggregation scheme without trusted authority in smart grid," *IEEE Systems Journal*, vol. 15, no. 1, pp. 395–406, 2020.

[48] H. Shen, Y. Liu, Z. Xia, and M. Zhang, "An efficient aggregation scheme resisting on malicious data mining attacks for smart grid," *Information Sciences*, vol. 526, no. 6, pp. 289–300, 2020.

[49] R. Fischer, M. Edward Halibozek, E. P. Halibozek, and D. Walters, *Introduction to security*. Butterworth-Heinemann, 2012.

[50] R. Lu, "Privacy-preserving multifunctional data aggregation," in *Privacy-Enhancing Aggregation Techniques for Smart Grid Communications*. Springer, 2016, pp. 85–110.

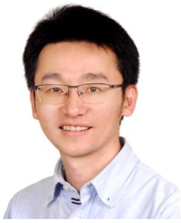


**Yuan Chang** received the master's degree from School of Computer Technology, Northeastern University, Shenyang, China, in 2022. He is working toward the Ph.D. degree with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China. His current research interests include cryptography, data security and privacy protection.

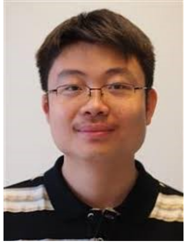
**Jiliang Li** received the Dr. rer. nat. degree in computer science from the University of Göttingen, Germany, in 2019. He is currently a Researcher Professor and PhD Supervisor with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China. His research interests include information security, cryptography, blockchain and IoT security.

**Ning Lu** received the B.Sc. degree in Information and Computing Science from Inner Mongolia University, Huhhot, China, in 2006, M.S. degree from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2009 and Ph.D. candidate in State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2013. Now, he is a associated professor in Northeastern University. His current research interests include artificial intelligence security, data security and privacy protection, Denial of Service attack defense.

**Wenbo Shi** received the M.S. degree from the Inha University, Incheon, South Korea, in 2007 and the Ph.D. degree from the Inha University, Incheon, South Korea, in 2010. Currently he is a professor at Northeastern University at Qinhuangdao. His research interests include cryptographic protocol, cloud computing security, artificial intelligence security, data security and privacy protection, Denial of Service attack defense.



**Zhou Su** has published technical papers, including top journals and top conferences, such as IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE/ACM TRANSACTIONS ON NETWORKING, and INFOCOM. His research interests include multimedia communication, wireless communication, and network traffic. Dr. Su received the Best Paper Award of International Conference IEEE ICC2020, IEEE BigdataSE2019, and IEEE CyberSciTech2017. He is an Associate Editor of IEEE INTERNET OF THINGS JOURNAL, IEEE OPEN JOURNAL OF COMPUTER SOCIETY, and IET Communications.



**Weizhi Meng** is currently an Associate Professor in the Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), Denmark. He obtained his Ph.D. degree in Computer Science from the City University of Hong Kong, Hong Kong. Prior to joining DTU, he worked as Research Scientist in Institute for Infocomm Research, A*Star, Singapore. He won the Outstanding Academic Performance Award during his doctoral study, and is a recipient of the Hong Kong Institution of Engineers (HKIE) Outstanding Paper Award for Young Engineers/Researchers in both 2014 and 2017. He also received the IEEE ComSoc Best Young Researcher Award for Europe, Middle East, & Africa Region (EMEA) in 2020. His primary research interests are cyber security and intelligent technology in security, including intrusion detection, IoT security, biometric authentication, and blockchain. He is senior member of IEEE.