Hindawi
Security and Communication Networks
Volume 2022, Article ID 4548374, 10 pages
https://doi.org/10.1155/2022/4548374

Research Article

A Traceable and Anonymous Data Aggregation Scheme with Fog Computing Devices for Smart Grid

Fan Wu [1] **and Xiong Li** [2]

[1]*School of Management, Xiamen University Tan Kah Kee College, Zhangzhou 363105, China*
[2]*Institute for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*

Correspondence should be addressed to Xiong Li; lixiongzhq@163.com

Received 22 February 2022; Revised 3 April 2022; Accepted 8 April 2022; Published 22 June 2022

Academic Editor: AnMin Fu

As an extension of the Internet of Things (IoT), smart city is a new aim for applications used in different industrial aspects. Intelligent IoT devices are used everywhere in smart cities to implement the functions such as monitoring and managing. Smart grid is one of the critical parts. Obtaining the quantity and cost of electricity usage is the most critical task of the smart grid. But such data in plaintext may leak the user privacy. So, it is an emergent aim to protect the private information of the user. Thus, we present a secure scheme for the smart grid, which not only protects the user's information including identity and power consumption in the communications but also tracks the correct electricity cost for each user. Also, electricity consumption data from users could be aggregated in fog devices and then analyzed by the utility provider. The formal proof shows that the message transmission reaches the security level of the chosen plaintext attack (CPA). Also, the security properties of the scheme express the robustness. Finally, the performance study demonstrates that the proposed protocol is acceptable in practice.

1. Introduction

Nowadays, smart city aims to optimize every function of the city and improve the quality of citizens' life with data analysis and technology development. Quality of life has been improved substantially in smart city. To complete the whole process from data collection to instruction assignment to concrete sensors, intelligent Internet of Things (IoT) devices such as smart meters, robots, aggregation devices, and software-defined production processes are deployed for different sorts of usage. Through sensors distributed everywhere, many kinds of data are gathered and sent to center servers for decision-making. Under that background, power grid, traffic, agriculture, and accommodation turn to be smart grid, smart traffic, smart agriculture, and smart home, respectively. However, security gaps containing weak authentication or vulnerability in code pieces lead to serious and urgent risks. Attackers can exploit the operating system and software's holes, eavesdrop on the messages in public channels between sensors and servers, or compromise sensors to get valuable information after analyzing relative data flows.

Traditional electricity grid consists of power stations, high-voltage transmission lines, and distribution lines. They build a large network that delivers electricity from the producer to the users, and the balance of power supply and usage relies on the construction of power plants. The more electricity is required, the more plants should be built. But with the technology progressing, smart grid becomes an attractive term, and it turns to be true in some developed countries. Promising changes appear in every aspect, including intelligent power generation, transmission, and applications. Advanced metering infrastructure (AMI) is the architecture of smart grid, where bidirection demand response is an important property, which permits the users and the utility provider (UP) to monitor, adjust, count, and forecast the electricity use. The time-of-use pricing mechanism is employed, and users should pay higher fees in the peak time under this measure. Also, smart grid is one of the most important sorts of cyber-physical systems (CPSs). As

physical devices, smart meters (SMs) are distributed widely outside the houses or gathered in a meter box in a building. Each smart meter collects the power consumption value in one house with every fixed period to calculate the fees due to different prices in peaks and valleys. Since the power usage data are fine-grained, smart grid faces security problems containing threats and weaknesses including identity and consumption leaking, replay attack, denial-of-service attack, and so forth. The attacker may deduce user habits or behaviors via such information, so the privacy of the personal information turns to be an important issue and is discussed by researchers. According to [1], several requirements should be satisfied, including transmitted data privacy, data reliability, and authentication between participants.

With the popularization of smart grid, enormous data have been generated. Simple data processing is done on fog devices (FDs). Fog computing means that some critical computations are completed on the edge of the network or fog devices distributed everywhere. SMs in one domain submit their collected data to the corresponding FD, and the extensional calculations, like clarifying the consumption fee and making data aggregations, could be done based on the data owned by FDs. All transmitted messages should be kept away from danger. Authentication and public key mechanism [2–4] are the necessary ways to protect the security of data. It is necessary to study the current situation of privacy-preserving schemes for smart grid and we will list the related literature.

1.1. Related Work. In recent years, a host of schemes for AMI has been presented. Based on [5], there are three classic sorts: key agreement, only consumption data encryption, and data aggregation. Authentication and key agreement is the usual way for sending information in smart grid [2–4, 6–15]. In 2011, Fouda et al. [6] presented a key agreement scheme between the building area network and home area network. But, it is criticized in [7] that heavy computation cost is used. Then, Li et al. [11] pointed out that identities of users were exposed by plaintext in [7]. In 2016, Tsai and Lo [8] presented a scheme with a session key formed between the smart meters and corresponding service supporter. But Odelu et al. and He et al. [9, 10] considered that calculation time in [8] costs too much due to bilinear pairing calculations. Unfortunately, the two schemes could not satisfy the anonymity of user [2]. In 2018, schemes [12–14] were proposed, but the weaknesses like lack of forward security and anonymity were still troubling people.

The second sort is that only the consumption data are protected, but the user's identity is not considered as the secret [5, 16–20]. In 2015, Diao et al. [16] proposed a scheme built on zero-knowledge and Camenisch-Lysyanskaya signature. They claimed that user identity was anonymous and linkable in the scheme, but soon, the forgery attack on the scheme was given in [17]. In 2016, Sui et al. [18] designed a scheme between SM and the electricity utility. But the user who consumes more electricity than the threshold will be exposed in plaintext. Next year, Ge et al. [19] pointed out that unlinkability feature could not be satisfied in [18]. But the two schemes [18, 19] are unfit due to exposing user identity simply and crudely. And in [5], secure channel is required several times when normal data transmission proceeds, and the identity is also in plaintext against the anonymity requirement. In 2020, Ding et al. [20] put forward a data aggregation scheme for smart grid, but both the identity and public key of user were transmitted directly in the public channel. In 2021, Su et al. [21] proposed a changeable threshold-based aggregation scheme for smart grid. However, identity is ignored in the message transmission, and only the aggregation value can be obtained in the control center, which is equal to UP here. Also, Wang et al. [22] presented an aggregation scheme keeping privacy of user in the same year, but the identity of user was still ignored.

The last sort is aggregating data relative to the consumption [1, 23–29]. In 2017, He et al. proposed studies [23, 24] which described the aggregation between SMs and the special aggregator. But in the aggregation part, the identity of user, which should be transmitted in plaintext, is needed to verify the data. Wang's scheme [25], which employed the identity-based signature, had the same problem, where the user's identity must be exposed in the public channel or the final check on the aggregation device could not be completed. In 2017, Badra and Zeadally [1] presented a scheme with symmetric homomorphic encryption and Diffie–Hellman problem. Every time, one user should update the shared key between the server and himself, with the help of another user. But how to find the suitable helper is not mentioned. Shen et al. [26] proposed a cube-data aggregation scheme for smart grid. However, the user identity is also in plaintext. Lu et al. [27] proposed a Paillier encryption-based scheme to make the data aggregation. But the time-based hash chain in the scheme is not suitable if the smart meter malfunctions once. The fog device cannot check the correct submission, while the last one or several submissions are lost or rejected. Liu et al. [28] used lifted EC-ElGamal cryptosystem with plaintext identity. However, except [1], all of the above schemes do not consider the fee of each user. Only aggregation and some statistical operations, such as mean and variance values, are regarded. Smart grid is first for electrical consumption, and the fee of electricity usage is much more important than statistical data for prognosis. On this aspect, in 2016, Wang et al. [29] proposed a scheme which could not only disclose the user consumption but also collect statistical data for computing statistic values. However, some weaknesses are exposed: the data in the aggregation process cannot be verified, the adversary can calculate the private consumption from the message, and the pseudoidentity will be exhaustively searched on the trusted server side by doing both hash and scalar multiplication, since the trusted server only calculates the collected data of the special smart meter which is required by UP.

Usually, the quantity of electricity usage is set as the discrete logarithm in aggregation. Based on [30], the power energy consumption in China is about 7225.5 terawatt/hour in 2017. This number is at the level of 2^{48}, and such discrete logarithm could be solved in 0.1 s [31], generally with the

Table 1: Notations.

Symbol	Meaning
SM_i, ID_{SM_i}	i^{th} smart meter and its identity
FD_j, ID_{FD_j}	j^{th} fog device and its identity
TS	Trusted server
UP, ID_{UP}	Utility provider and its identity
G	A cyclic group on a finite field F_n from an elliptic curve
q	A large prime, which is the order of G
P	Generator of G
Z_q^*	Multiplicative group modulo q
$h_i (i = 1, 2, \ldots 10)$	Hash functions
$\oplus$	Exclusive or operation
$\|$	Concatenation operation
V	Distribution of power consumption in a domain
v_i	Quantity of usage on SM_i
l_s	Security length
$z_j/Z_j = z_jP$	Private/public key for FD_j
$x/X = xP$	Private/public key for TS
$y/Y = yP$	Private/public key for UP
key_j	The common secret key between FD_j and UP
$\mathscr{A}$	The adversary
M_1, M_2, M_3	Messages

Pollard rho algorithm [32]. This technology is employed in many studies [23–25, 27, 29, 33].

1.2. Contributions.

(1) We give a new data transmission scheme for smart grid, and it could make both the power consumption and data aggregation clear.

(2) Formal proof demonstrates that the messages are robust enough against forgery attacks

(3) Considering the security characters and performance evaluation, our scheme is good for practicality

1.3. Organization of the Paper. The rest of study is organized as follows: Section 2 expresses the basic knowledge of the study. Our scheme is in Section 3. Then, the formal proof lies in Section 4 and security analysis is in Section 5. The performance situation is in Section 6. Finally, the conclusion is drawn in Section 7.

2. Preliminary

2.1. Notations. In Table 1, the notations used throughout the study are given.

2.2. Referred Mathematical Problems. The problems given in the following are based on the elliptic group G with order q and generator P mentioned in Table 1.

Definition 1. The discrete logarithm (DL) problem is that in the tuple (P, aP), where $a \in Z_q^*$ is unknown, it is hard to calculate a.

Definition 2. The computational Diffie–Hellman (CDH) problem on G is that given the tuple $(\mathrm{P}, \mathrm{aP}, \mathrm{bP})$, where $a, b \in Z_q^*$ are unknown, it is hard to calculate $abP \in G$.

Definition 3. The decisional Diffie–Hellman (DDH) problem on G is that given the tuple (P, aP, bP, cP), where $a, b \in Z_q^*$ are unknown, it is hard to judge if $cP = abP$.

Definition 4. The gap Diffie–Hellman (GDH) problem on G is that given tuple (P, aP, bP), where $a, b \in Z_q^*$ are unknown, it is hard to calculate $abP \in G$ via the help of DDH tool, like an oracle. Here, we define $\varepsilon = \mathrm{Adv}_{\mathscr{A}}^{\mathrm{GDH}}(t)$, which is the probability of solving GDH problem for $\mathscr{A}$ in polynomial time t.

2.3. Network Model. Our scheme relies on the model in [29]. If the users obey the laws and pay the bill in time, it is unnecessary to expose the fine-grain consumption value to smart grid, especially UP. UP only accepts the aggregation of consumption and changes power supply with the corresponding price in different time spans. The architecture of the network is shown in Figure 1. Four kinds of devices are in the network: smart meter, fog device, trusted server (TS), and utility provider. Smart meter measures the power consumption details for every house. It is natural that the fee should be checked. But in [29], the fee is calculated on SM side. Without verifications on some trusted server, such data cannot be believed. Different from [29], only the consumptions will be submitted. Fees are not considered. Also, only total electricity consumption in one house is submitted in our scheme. Consumptions from all appliances in the house are not considered. Such data are submitted to the fog device, which stores the collection and is ready to provide data for people to check the consumption and to aggregate the data without leaking user information. The trusted server can get power consumption from each user, in order to calculate the electricity fee later. Also, it generates the key for the fog device to send the aggregated data with encryption, in order to prevent the attacker from cracking. The utility provider requires the aggregated statistical data to predict the total data in a long time. Wired channels are between fog devices and the trusted server and between fog devices and UP.

2.4. Security Model. Combining studies [27, 29, 34], we give the security model of our scheme as follows.

(1) The scheme is chosen plaintext attack (CPA) secure

(2) The trusted server is reliable and can get the real identity and power consumption of user. It is for the property of user anonymity and traceability.

(3) $\mathscr{A}$ could eavesdrop and forge messages in the public channel

(4) The UP and fog devices are honest-but-curious, since they execute the protocol but are curious about the privacy of users. However, according to [27], any FD_j will not collude with UP.

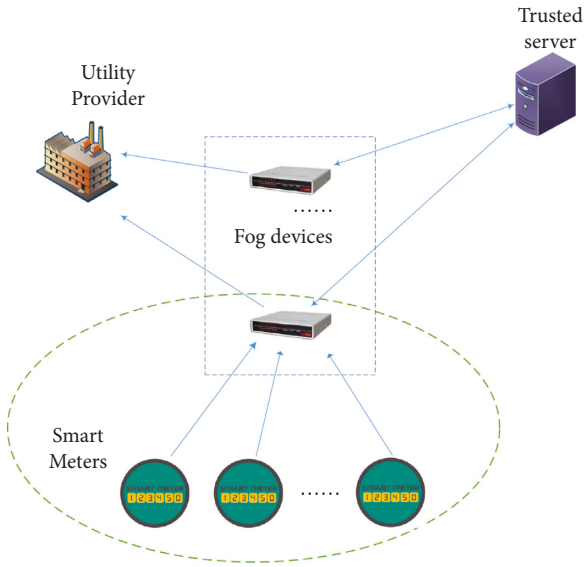

Figure 1: Network model.

3. Proposed Scheme

We divide our scheme into six phases: initialization, data encryption, consumption affirmation, aggregation key generation, aggregation, and aggregated data decryption. Different from [29], we do not consider user's fee submission, as the fee should be checked on the TS side, and it is not suitable to completely trust the fee calculated by smart meters. Moreover, we focus on the unitary consumption or one smart meter for a house. We do not use the way in [29], where each appliance is counted, respectively. Moreover, there is only one TS and one UP. Moreover, we put the process from Section 3.2 to Section 3.6 in Table 2.

3.1. Initialization. TS generates a cyclic group G with a large prime order q and generator P, as given in Table 1. Its private/public key pair is $(x/X = xP)$. Similarly, UP owns its private/public key pair $(y/Y = yP)$, and FD_j owns its private/public key pair $(z_j/Z_j = z_jP)$. Moreover, UP's identity is $\mathrm{ID}_{\mathrm{UP}}$ and FD_j's identity is $\mathrm{ID}_{\mathrm{FD}_j}$; TS stores the pair $(\mathrm{ID}_{\mathrm{SM}_i}, \mathrm{ID}_{\mathrm{FD}_j})$, where the submission target of $\mathrm{ID}_{\mathrm{SM}_i}$ is $\mathrm{ID}_{\mathrm{FD}_j}$; SM_i also stores the pair $(\mathrm{ID}_{\mathrm{SM}_i}, \mathrm{ID}_{\mathrm{FD}_j})$, such that SM_i should submit the information to FD_j. At last, FD_j and UP have a common secret key key_j. UP stores the quantity of smart meters N_j in FD_j's domain. Finally, hash functions are defined as follows: $h_i\,(i = 1, 2, 3, 5, 6, 7, 8, 9, 10)$: $\{0,1\}^* \mapsto \{0,1\}^{l_s}$ and h_4: $\{0,1\}^* \mapsto Z_q^*$.

3.2. Data Encryption. SM_i selects random numbers $r_1, r_2 \in Z_q^*$, picks the timestamp t_1, and calculates the following elements with user consumption v_i: $A_{i,1} = r_1P$, $A_{i,2} = r_2P$, $A_{i,3} = h_1(r_1X) \oplus v_i$, $A_{i,4} = h_2(r_2X) \oplus \mathrm{ID}_{SM_i}$, $A_{i,5} = 2r_1X + r_2X + v_iP$, $w_i = v_i^2$, $A_{i,6} = r_1X + 2r_2X + w_iP$, and $A_{i,7} = h_3(r_1X \| v_i \| ID_{SM_i} \| ID_{FD_j} \| w_i \| r_2X \| t_1)$. Then, SM_i sends $M_1 = \{A_{i,1}, A_{i,2}, A_{i,3}, A_{i,4}, A_{i,5}, A_{i,6}, A_{i,7}, t_1\}$ to the corresponding FD_j.

3.3. Consumption Affirmation. FD_j checks t_1 and stores M_1 if t_1 is valid. Then, it sends M_1 to TS, in order to disclose the user identity and corresponding power consumption. TS computes $B_1 = xA_{i,1}$, $B_2 = xA_{i,2}$, $v_i = A_{i,3} \oplus h_1(B_1)$, $\mathrm{ID}_{SM_i} = A_{i,4} \oplus h_2(B_2)$, $B_3 = A_{i,5} - 2B_1 - B_2$, and $B_4 = A_{i,6} - B_1 - 2B_2$, finds out the corresponding FD_j according to SM_i, and checks if $B_3 = v_iP$, $w_i = v_i^2$, $B_4 = w_iP$, and $A_{i,7} = h_3(B_1 \| v_i \| ID_{SM_i} \| ID_{FD_j} \| w_i \| B_2 \| t_1)$. If true, TS could calculate the fee according to v_i.

3.4. Aggregation Key Generation. In order to help FD_j aggregate power usage in smart meters for a time span, TS selects a random string s, the timestamp t_2 in relative time span and $r_3 \in Z_q^*$, and calculates $B_5 = r_3P$, $B_6 = xh_4(r_3Y \| s) \oplus h_5(r_3Z_j \| ID_{FD_j})$, $B_7 = h_4^{-1}(r_3Y \| s) \oplus h_6(r_3Y \| ID_{UP})$, $B_8 = h_7(xh_4(r_3Y \| s) \| ID_{FD_j} \| B_7 \| t_2)$, and $B_9 = h_8(h_4^{-1}(r_3Y \| s) \| r_3Y)$. Then, TS sends $M_2 = \{B_5, B_6, B_7, B_8, B_9, t_2\}$ to FD_j.

3.5. Aggregation. FD_j checks t_2, gets the number k as the number of functioning smart meters, picks up timestamp t_3 and $r_4 \in Z_q^*$, calculates $C_1 = B_6 \oplus h_5(z_jB_5 \| ID_{FD_j})$, and checks if $B_8 = h_7(C_1 \| ID_{FD_j} \| B_7 \| t_2)$. If true, FD_j computes $C_2 = \sum_{i=1}^{k} A_{i,1}$, $C_3 = C_1C_2$, $C_4 = \sum_{i=1}^{k} A_{i,2}$, $C_5 = C_1C_4$, $C_6 = B_5$, $C_7 = B_7$, $C_8 = \sum_{i=1}^{k} A_{i,5}$, $C_9 = \sum_{i=1}^{k} A_{i,6}$, $C_{10} = B_9$, $C_{11} = r_4P$, $C_{12} = h_9(r_4Y \| t_3) \oplus k$, and $C_{13} = h_{10}(C_3 \| C_5 \| C_7 \| C_8 \| C_9 \| C_{10} \| k \| key_j \| t_3)$. Then, FD_j sends $M_3 = \{C_3, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11}, C_{12}, C_{13}, t_3, ID_{FD_j}\}$ to UP.

3.6. Aggregated Data Decryption. UP checks t_3 and searches key_j based on $\mathrm{ID}_{\mathrm{FD}_j}$. If the checks and the data are found, it computes $k = C_{12} \oplus h_9(yC_{11} \| t_3)$ and checks $c_{13} = \{C_3, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11}, C_{12}, C_{13}, t_3, ID_{FD_j}\}$. If true, UP computes $D_1 = C_7 \oplus h_6(yC_6 \| ID_{UP})$ and checks if $C_{10} = h_8(D_1 \| yC_6)$. If correct, UP computes $D_2 = C_8 - 2D_1C_3 - D_1C_5$ and $D_3 = C_9 - D_1C_3 - 2D_1C_5$ and then uses the Pollard rho algorithm to get $W_1 = \sum_{i=1}^{k} v_i$ from D_2 and $W_2 = \sum_{i=1}^{k} w_i$ from D_3. If $k = N_j$, the mean value $E(V) = W_1/k$ and the variance value $\mathrm{Var}(V) = W_2/k - W_1^2/k^2$; else, if $k < N_j$, the variance value is changed to be $\mathrm{Var}(V) = 1/(k-1)(W_2 - W_1^2/k)$.

4. Formal Proof

Nowadays, researchers consider that attackers should have negligible probability to retrieve any plaintext from ciphertext in cryptographic protocols. Such protocol should meet indistinguishability (IND) security which means that $\mathscr{A}$ could not distinguish two plaintexts, while one of corresponding ciphertexts is given. In this study, chosen plaintext attack (CPA), which means $\mathscr{A}$ does not have decryption right for any ciphertext he selects, will be proved for our scheme.

Our scheme is under IND-CPA secure. The concrete proof is given.

4.1. Basic Knowledge of IND-CPA Security for Our Scheme. Three games are brought in to explain the security for the three messages. We show the game process and then give the

Table 2: Data flow for consumption submission.

SM_i	FD_j	TS	UP
Select r_1, r_2, t_1 $A_{i,1} = r_1 P$ $A_{i,2} = r_2 P$, $A_{i,3} = h_1(r_1 X) \oplus v_i$ $A_{i,4} = h_2(r_2 X) \oplus ID_{SM_i}$ $A_{i,5} = 2r_1 X + r_2 X + v_i P$ $w_i = v_i^2$ $A_{i,6} = r_1 X + 2r_2 X + w_i P$ $A_{i,7} = h_3(r_1 X \Vert v_i \Vert ID_{SM_i} \Vert ID_{FD_j} \Vert w_i \Vert r_2 X \Vert t_1)$ $\longrightarrow^{M_1} = (A_{i,1}, A_{i,2}, A_{i,3}, A_{i,4}, A_{i,5}, A_{i,6}, A_{i,7}, t_1)$			
	Check t_1, stores M_1 $\longrightarrow^{M_1}$		
		$B_1 = xA_{i,1}$, $B_2 = xA_{i,2}$ $v_i = A_{i,3} \oplus h_1(B_1)$ $ID_{SM_i} = A_{i,4} \oplus h_2(B_2)$ $B_3 = A_{i,5} - 2B_1 - B_2$, $B_4 = A_{i,6} - B_1 - 2B_2$ Get FD_j Check $B_3 ? = v_i P$, $w_i ? = v_i^2$, $B_4 ? = w_i P$ And $A_{i,7} ? = h_3(B_1 \Vert v_i \Vert ID_{SM_i} \Vert ID_{FD_j} \Vert w_i \Vert B_2 \Vert t_1)$ Select s, t_2, r_3 $B_5 = r_3 P$ $B_6 = xh_4(r_3 Y \Vert s) \oplus h_5(r_3 Z_j \Vert ID_{FD_j})$ $B_7 = h_4^{-1}(r_3 Y \Vert s) \oplus h_6(r_3 Y \Vert ID_{UP})$ $B_8 = h_7(xh_4(r_3 Y \Vert s) \Vert ID_{FD_j} \Vert B_7 \Vert t_2)$ $B_9 = h_8(h_4^{-1}(r_3 Y \Vert s) \Vert r_3 Y)$ $\overset{M_2=(B_5,B_6,B_7,B_8,B_9,t_2)}{\longleftarrow}$	
	Check t_2, get k, t_3, r_4 $C_1 = B_6 \oplus h_5(z_j B_5 \Vert ID_{FD_j})$ Check $B_8 ? = h_7(C_1 \Vert ID_{FD_j} \Vert B_7 \Vert t_2)$ Compute $C_2 = \sum_{i=1}^{K} A_{i,1}$, $C_3 = C_1 C_2$ $C_4 = \sum_{i=1}^{k} A_{i,2}$, $C_5 = C_1 C_4$ $C_6 = B_5$, $C_7 = B_7$, $C_8 = \sum_{i=1}^{k} A_{i,5}$ $C_9 = \sum_{i=1}^{k} A_{i,6}$, $C_{10} = B_9$ $C_{11} = r_4 P$, $C_{12} = h_9(r_4 Y \Vert t_3) \oplus k$ $C_{13} = h_{10}(C_3 \Vert C_5 \Vert C_7 \Vert C_8 \Vert C_9 \Vert C_{10} \Vert k \Vert key_j \Vert t_3)$ $M_3 = \overset{(C_3,C_5,C_6,C_7,C_8,C_9,C_{10},C_{11},C_{12},C_{13},t_3,ID_{FD_j})}{\longleftarrow}$ $C_{12}, C_{13}, t_3, ID_{FD_j}\}$		
			Check t_3, search key_j Compute $k = C_{12} \oplus h_9(yC_{11} \Vert t_3)$ Check $C_{13} ? = h_{10}(C_3 \Vert C_5 \Vert C_7 \Vert C_8 \Vert C_9 \Vert C_{10} \Vert k \Vert key_j \Vert t_3)$ Compute $D_1 = C_7 \oplus h_6(yC_6 \Vert ID_{UP})$ Check $C_{10} ? = h_8(D_1 \Vert yC_6)$ Compute $D_2 = C_8 - 2D_1 C_3 - D_1 C_5$ $D_3 = C_9 - D_1 C_3 - 2D_1 C_5$ get $W_1 = \sum_{i=1}^{k} v_i$ from D_2 get $W_2 = \sum_{i=1}^{k} w_i$ from D_3 Compute the mean value and variance value according to k

analysis of the games. The results show how the scheme keeps IND-CPA security. The proposed scheme meets the CPA security requirements if the polynomial time adversary $\mathcal{A}$ has negligible probability to win the games. A simulator $\mathcal{S}$ is used to provide the random oracle query service and $\mathcal{A}$ makes queries to try to break the IND-CPA security. All games can be divided into five phases as follows:

(1) Initialization: $\mathcal{S}$ generates system parameters including G with generator P and large prime order q, public keys of FD_j, UP, and TS, identities of smart meters, fog devices, trusted server and utility provider, secret keys between FD_j and UP, and hash functions. We define the public keys of TS. UP and FD_j are $X = a_{\mathrm{TS}}P$, $Y = a_{\mathrm{UP}}P$, and $Z = a_jP$, respectively, where a_{TS}, a_{UP}, and a_j are unknown.

(2) Query 1: $\mathcal{A}$ queries the hash oracles, and $\mathcal{S}$ returns the results.

(3) Challenge: $\mathcal{A}$ selects fresh information $info^0$ and $info^1$ to $\mathcal{S}$. $\mathcal{S}$ then chooses a bit $\omega \in \{0, 1\}$. And $info^\omega$ is used to generate the corresponding message in the game.

(4) Query 2: it is same as query 1.

(5) Guess: $\mathcal{A}$ should give a bit ω' as the result of guessing ω. If $\omega = \omega'$, $\mathcal{A}$ wins the game.

$\mathcal{A}$ knows all public parameters and all identities of participants. He will ask h_i oracle for $q_{h_i}(i = 1, 2, \ldots 10)$ times. There are hash lists for storing $\mathcal{A}$'s corresponding hash queries. For instance, L_{h_i} stores $(i, str, result)$, where h_i is queried by $\mathcal{A}$ and $result$ is the hash result of str. The advantage for $\mathcal{A}$ breaking the message $M_b(b = 1, 2, 3)$ is denoted as $A\,dv_{M_b}^{IN\ D-CPA}(\mathcal{A}) = \Pr[\omega = \omega'] - 1/2$. In the following proofs, we only consider the extra probabilities beyond 1/2. As mentioned in Section 2.2, $\varepsilon = A\,dv_{\mathcal{A}}^{G\ DH}(t)$, and we employ the probability to express the hard level of solving the GDH problem in the following theorems. That probability is used when we find the tuple format (aP, bP, abP) occurring in the hash list L_{h_i}, where a and b are unknown, as we mentioned also in Section 2.2.

4.2. Proof of CPA

Theorem 1. *The data encryption phase is IND-CPA secure and* $A\,dv_{M_1}^{IN\ D-CPA}(\mathcal{A}) \le (q_{h_1}^2 + q_{h_2}^2 + q_{h_3}^2 + 2(q_{h_1} + q_{h_2} + q_{h_3}))/2^{l_s+1} + q_{h_1}q_{h_2}q_{h_3}\varepsilon^2$.

Proof. The concrete operations are as follows:

(1) Initialization: $\mathcal{S}$ produces parameters as mentioned in Section 4.1.

(2) Query 1: $\mathcal{A}$ makes h_1, h_2, and h_3 queries with the string str. $\mathcal{S}$ searches if there is the existed queried tuple $(i, str, result)$ in $L_{h_i}(i = 1, 2, 3)$. If true, $result$ will be returned. Otherwise, $\mathcal{S}$ selects $result \in \{0, 1\}^{l_s}$, and the tuple $(i, str, result)$ is written into the list.

(3) Challenge: $\mathcal{A}$ selects v_i^0 and v_i^1 and submits them to $\mathcal{S}$. $\mathcal{S}$ produces M_1^ω as the following operations. First, a bit ω is chosen and v_i^ω is used to produce the message.

Then, the corresponding message is sent to $\mathcal{A}$.

(4) Query 2: $\mathcal{A}$ makes h_1, h_2, and h_3 queries again until the numbers $q_{h_i}(i = 1, 2, 3)$ are reached.

(5) Guessing: $\mathcal{A}$ gives a bit ω'.

If $\omega = \omega'$, we divide the advantage of $\mathcal{A}$'s guessing into three parts. First, to avoid the collision of hash results, the upper probability is $(q_{h_1}^2 + q_{h_2}^2 + q_{h_3}^2)/2^{l_s+1}$, based on birthday paradox. Second, if the hash results are guessed correctly without oracle queries, the probability is at most $(q_{h_1} + q_{h_2} + q_{h_3})/2^{l_s}$. Finally, if $\mathcal{A}$ could judge the message by generating a correct one for comparison, $(1, r_1X, *)$, $(2, r_2X, *)$, and $(3, r_1X \| * \| r_2X \| *, A_{i,7})$ can be obtained from L_{h_1}, L_{h_2}, and L_{h_3}, respectively. So, the condition of $\mathcal{A}$ querying the correct strings is solving both GDH problems, and the probability is $q_{h_1}q_{h_2}q_{h_3}\varepsilon^2$. Then, the theorem is deduced. □

Theorem 2. *The aggregation key generation phase is IND-CPA secure and* $A\,dv_{M_2}^{IN\ D-CPA}(\mathcal{A}) \le (q_{h_4}^2 + 2q_{h_4})/2(q-1) + (q_{h_5}^2 + q_{h_6}^2 + q_{h_7}^2 + q_{h_8}^2 + 2(q_{h_5} + q_{h_6} + q_{h_7} + q_{h_8}))/2^{l_s+1} + q_{h_4}q_{h_5}q_{h_6}q_{h_8}\varepsilon^2$.

Proof. The concrete operations are as follows:

(1) Initialization: $\mathcal{S}$ produces parameters as mentioned in Section 4.1.

(2) Query 1: $\mathcal{A}$ makes h_4, h_5, h_6, h_7, and h_8 queries with the string str. $\mathcal{S}$ searches if there is the existed queried tuple $(i, str, result)$ in $L_{h_i}(i = 4, 5, 6, 7, 8)$. If true, $result$ will be returned. Otherwise, $\mathcal{S}$ selects $result \in \{0, 1\}^{l_s}(i = 5, 6, 7, 8)$ or $result \in Z_q^*(i = 4)$, and the tuple $(i, str, result)$ is written into the list.

(3) Challenge: $\mathcal{A}$ selects strings s^0 and s^1 and sends them to $\mathcal{S}$. Then, M_2^ω is produced as follows. First, $\mathcal{S}$ chooses a bit ω and s^ω is used to produce the message M_2^ω.

Then, the corresponding message is sent to $\mathcal{A}$.

(4) Query 2: $\mathcal{A}$ makes h_4, h_5, h_6, h_7, and h_8 queries again until the numbers $q_{h_i}(i = 4, 5, 6, 7, 8)$ are reached.

(5) Guessing: $\mathcal{A}$ gives a bit ω'.

If $\omega = \omega'$, we divide the advantage of $\mathcal{A}$'s guessing into three parts. First, to avoid the collision of hash results, the upper probability is $q_{h_4}^2/2(q-1) + (q_{h_5}^2 + q_{h_6}^2 + q_{h_7}^2 + q_{h_8}^2)/2^{l_s+1}$, based on birthday paradox. Second, if the hash results are guessed correctly without oracle queries, the probability is at most $q_{h_4}/q - 1 + (q_{h_5} + q_{h_6} + q_{h_7} + q_{h_8})/2^{l_s}$. Finally, if $\mathcal{A}$ could judge the message by generating a correct one for comparison, $(4, r_3Y \| *, *) \in L_{h_4}$, $(5, r_3Z_j \| *, *) \in L_{h_5}$, $(6, r_3Y \| *, *) \in L_{h_6}$, and $(8, * \| r_3Y, *) \in L_{h_8}$ could be found. The probability is at least $1/q_{h_i}(i = 4, 5, 6, 8)$. Like the analysis in Theorem 1, the probability is $q_{h_4}q_{h_5}q_{h_6}q_{h_8}\varepsilon^2$. So, we get the theorem. □

Theorem 3. *The aggregation phase is IND-CPA secure and* $Adv_{M_3}^{IN\ D-CPA}(\mathcal{A}) \le q_{h_4}^2 + 2q_{h_4}/2(q-1) + (q_{h_6^2} + q_{h_8^2} + q_{h_9}^2 + q_{h_{10}}^2 + 2(q_{h_6} + q_{h_8} + q_{h_9} + q_{h_{10}}))/2^{l_s+1} + q_{h_4}/q - 1 + q_{h_4}q_{h_6}q_{h_8}\ q_{h_9}\varepsilon^2$.

Proof. The concrete operations are as follows:

(1) Initialization: $\mathcal{S}$ produces parameters as mentioned in Section 4.1.

(2) Query 1: according to M_3, all related hash functions $h_i\,(i = 4, 6, 8, 9, 10)$ are queried by $\mathcal{A}$. $\mathcal{S}$ searches if there is the existed queried tuple $(i, str, result)$ in $L_{h_i}\,(i = 4, 6, 8, 9, 10)$. If true, *result* will be returned. Otherwise, $\mathcal{S}$ selects $result \in \{0,1\}^{l_s}\,(i = 6, 8, 9, 10)$ or $result \in Z_q^*\,(i = 4)$, and the tuple $(i, str, result)$ is written into the list.

(3) Challenge: $\mathcal{A}$ selects v_i^0 and v_i^1 and submits them to $\mathcal{S}$. Then, M_3 is produced as follows: first, a bit ω is chosen, and s^ω is used to produce M_3^ω.

(4) Query 2: $\mathcal{A}$ makes h_4, h_6, h_8, h_9, and h_{10} queries again until the numbers $q_{h_i}\,(i = 4, 6, 8, 9, 10)$ are reached.

(5) Guessing: $\mathcal{A}$ gives a bit ω'.

If $\omega = \omega'$, we divide the advantage of $\mathcal{A}$'s guessing into three parts. First, to avoid the collision of hash results, the upper probability is $q_{h_4}^2/2(q-1) + (q_{h_6^2} + q_{h_8^2} + q_{h_9}^2 + q_{h_{10}}^2)/2^{l_s+1}$. Second, if the hash results are guessed correctly without oracle queries, the probability is at most $q_{h_4}/q - 1 + (q_{h_6} + q_{h_8} + q_{h_9} + q_{h_{10}})/2^{l_s}$. Finally, if $\mathcal{A}$ could judge the message by generating a correct one for comparison, $(4, r_3Y \| *, *) \in L_{h_4}$, $(6, r_3Y \| *, *) \in L_{h_6}$, $(8, * \| r_3Y, *) \in L_{h_8}$, and $(9, r_4Y \| *, *) \in L_{h_9}$ could be found. The probability is at least $1/q_{h_i}\,(i = 4, 6, 8, 9)$. Like the analysis in Theorem 1, the probability is $q_{h_4}q_{h_6}q_{h_8}q_{h_9}\varepsilon^2$. So, we get the theorem. □

5. Security Property Expression

The security properties are illustrated, and we compare our scheme with some recent ones [18, 20, 22, 27, 29]. Readers may search for some concrete details in corresponding studies. The results are given in Table 3. ✓ denotes the scheme meets the security property, while × denotes the opposite case. If the property is not fit for the scheme, ∅ is used. P1–P7 denote confidentiality, user anonymity, traceability, data aggregation, scalability, against internal attacks, and replay attacks, respectively. From the results, we see that the proposed scheme meets all the security requirements.

5.1. Confidentiality. First, we discuss our scheme. For M_1, if $\mathcal{A}$ wants to get v_i and ID_{SM_i}, he should know TS's secret key x to get r_1X and r_2X from r_1P and r_2P. For M_2, if $\mathcal{A}$ wants to get the critical element $xh_4(r_3Y||s)$, he must know any of the private keys y or z_j. For M_3, if $\mathcal{A}$ wants to get the element $h_4^{-1}(r_3Y||s)$, he must know any private keys same as in M_2.

Moreover, in [29], the keys for reencryption are directly sent to the public cloud server in the rekey phase, and they are exposed in the channel. Also, there is no authentication between the public cloud server and UP, so the data generated in this phase can be changed, e.g., adding P on c_2. UP cannot check the correctness of data. Last, the consumption information is leaked. In the Enc phase, power usage quan and the cost fee are sent by $rX + \mathrm{quan} \cdot P$ and $rX + \mathrm{fee} \cdot P$, respectively. $\mathcal{A}$ could calculate $\alpha = (\mathrm{fee} - \mathrm{quan})$ by subtraction and Pollard rho algorithm. Then, $\mathcal{A}$ gets $\alpha = \mathrm{quan}(\mathrm{fee}/\mathrm{quan} - 1)$. Fee/quan is just the price of electricity in the fixed period, and the power usage quan can be deduced.

Table 3: Security property.

	P1	P2	P3	P4	P5	P6	P7
Ours	✓	✓	✓	✓	✓	✓	✓
[29]	×	✓	✓	✓	×	✓	✓
[27]	✓	✓	×	✓	∅	✓	✓
[18]	×	×	✓	×	✓	∅	✓
[20]	✓	×	×	✓	∅	✓	✓
[22]	✓	∅	×	✓	∅	✓	✓

5.2. User Anonymity. In our scheme, $\mathrm{ID}_{\mathrm{SM}_i}$ is hidden by $h_2(r_2X)$. $\mathcal{A}$ should know TS's secret key x to calculate r_2X based on $A_{i,2} = r_2X$. But in [18], the user identity may be exposed. Information of users who consume more electricity than the threshold will be exposed in the channel, including identity and power consumption. Using more electricity is not a crime, and it is unsuitable to publish user information simply due to such a case. Also, in [20], user identity is in plaintext obviously. So, we use × for the two schemes. In [22], user identity is not needed in the entire scheme, and ∅ is used.

5.3. Trace Ability. We set Section 3.3 to make the power consumption of the user clear and to satisfy the basic function of the smart grid. TS computes $B_2 = xA_{i,2}$ and $ID_{SM_i} = A_{i,4} \oplus h_2(B_2)$ to get the identity $\mathrm{ID}_{\mathrm{SM}_i}$, accompanying with the consumption v_i. Such calculations can make TS get the (ID_{SM_i}, v_i) tuple and know the fee of the user consumption.

However, in [20, 22, 27], no entity except the smart meter itself knows the power consumption. How to affirm the user's fee for power usage is a difficult thing in the above three mentioned schemes.

5.4. Data Aggregation. Same as [20, 22, 27, 29], our scheme has the part of data aggregation, in order to analyze the statistical data on UP. But, in [18], data aggregation is not focused.

5.5. Scalability. In Section 3.3, TS does not require exhaustive searching for checking the identity. Or we say that no extra computation is before searching, even a hash result. But, in [29], if UP questions for some smart meter, the trusted server should use a scalar multiplication and a hash function to check all indexed users. We use × at the corresponding blank. But, such property does not fit for [20, 22, 27], since no tracking operation is in any of them.

5.6. Against Internal Attack. We make analysis based on the fourth item in Section 2.4. Since no fine-grained data appear in UP, if $\mathscr{A}$ colludes with UP to crack the messages, $\mathscr{A}$ still faces DL problem and GDH problem to get the timely private data from the messages due to lack of the private key x. A similar situation for colluding with fog devices can be deduced.

5.7. Against Replay Attack. To avoid replay attack, timestamps t_1, t_2, and t_3 are used in our scheme. Once $\mathscr{A}$ tries to modify any message, he must change the element for checking. In M_1, $A_{i,7}$ contains t_1, where r_1X and r_2X are also included. Computing the two results mean that two GDH problems should be cracked, based on $A_{i,1} = r_1P$, $A_{i,2} = r_2P$, and the public key of TS. In M_2, t_2 is used in B_8, where r_3Y, s, and x are also referred. Besides guessing s and x, computing r_3Y also means that one GDH problem should be cracked, based on $B_5 = r_3P$ and the public key of UP. Similar situation occurs in M_3. t_3 is used in C_{13}, where r_4Y is needed to crack based on $C_{11} = r_4P$ and public key of UP.

6. Performance Evaluation

In this section, we compare our scheme with [20, 22, 29] via time cost and communication cost. The test platform is MIRACL Library under Ubuntu 20, with Intel(R) Core(TM) i5-9400H CPU 2.50 GHz and 16.0 GB memory. The length of points on the elliptic curve is 320 bits, where the order of the additive group is 160 bits long. The timestamps have 64 bits, while all identities of devices have 32 bits and the hash function is Sha2-256. We use AES as the symmetric encryption/decryption algorithm in [29]. The symbols of time cost are given in Table 4, and the time cost comparison is given in Table 5. We use PH1–PH5 to express phases including data encryption, consumption affirmation, aggregation key generation, aggregation, and aggregated data decryption. k is the number of smart meters belonging to one fog device, and m is the number of fog devices.

From Table 5, we see that our scheme costs less than [29] in PH2 and PH5. In PH1, we use two random numbers for scalar multiplications. Since the power usage and fee are relatively small numbers, it is probable to calculate the private consumption, as we have demonstrated in Section 5.1 for [29]. In PH3, we add the operations to protect the rekeys, while there is no such idea in [29]. In PH5, since each submission of FD_j will cost the same time, we only list one round calculation in UP. Our scheme costs only a little more than [29]. Here, we should claim that there is no phase like PH2 and PH3 in schemes [20, 22], so the corresponding blanks are empty. In PH5, only one Pollard rho algorithm is used in [20, 22], since the final result is aimed at the aggregation value of consumption that is different from our scheme and [29], which also have another target of the variance value. In PH4, as long as a natural condition $k > 2$ exists, we see that our scheme costs better than [20, 22, 29]. The verifications of the process are settled on TS for our scheme and [29], unlike [20, 22], such the verification is put on the media devices. The reason is that the cost of user consumption needs to be calculated and affirmed.

Table 4: Time cost for referred cryptographical operations.

Symbol	Meaning	Time (ms)
T_s	Time of one scalar multiplication on G	0.2898
T_{se}	Time of symmetric encryption/decryption	0.01368
T_{2e}	Time of double exponentiation in the group	0.3528
T_h	Time of Sha2-256	0.003128
T_i	Time of one inversion in a group	0.02673
T_m	Time of one multiplication in a group	0.00169
T_a	Time of one point addition on G	0.00184
T_{PR}	Time of one pollard rho algorithm	61.4667

We illustrate the concrete communication cost here. For our scheme, in the data encryption phase, M_1 has $320 * 4 + 256 * 3 + 64 = 2112$ bits, and there are totally 2112km bits in one period. In the consumption affirmation phase, all FDs submit the collected messages to TS, and the total information is the same as the last step. In the aggregation key generation phase, M_2 has $320 + 256 * 4 + 64 = 1408$ bits, and there are totally 1408m bits in one transmission. In the aggregation phase, M_3 has $320 * 6 + 256 * 4 + 32 + 64 * 2 = 3104$ bits, and there are totally 3104m bits. So, the whole communication cost is 4224 km + 4512 m bits. We evaluate the transmission situation. Generally, channels between smart meters and their corresponding fog devices are considered to be wireless. Suppose a normal building for residents, about 30 floors, and generally, it has less than 200 houses. Each smart meter submits its data in every 15 minutes [35]. In the 15-minute period, there are less than $2112 * 200 = 422400$ bits in total or less than 470 bps, that is, a very small data rate. Second, we consider wired transmission messages M_2 and M_3. Suppose there are 100000 fog devices to send the messages. The total data volume is $100000 * 4512 = 451200000 < 500$ Mbits. Note such volume is for 15 minutes, and the fiber can support 10 Gbps rate [36]. So, the communication cost in our scheme is practical.

On the other side, for scheme in [29], in the enc phase, the message has $256 + 32 + 256 + 320 * 3 + 64 = 1568$ bits and in total 1568km bits. In the TTP-Dec phase, the message has $256 + 64 + 1568 = 1888$ bits and in total 1888 km bits. In the rekey phase, the message has $160 * 2 = 320$ bits and in total 320m bits. In the LiAgg-ReEnc phase, the message has $320 * 4 + 160 = 1440$ bits and in total 1440m bits. So, the whole communication cost is 3456km + 1760m bits. All could see that our scheme costs more than the scheme in [29]. However, according to our analysis of Section 4 and Section 5, our scheme is CPA secure and meets common security properties. Moreover, in [29], only CPA security is proved only for Enc and LiAgg-ReEnc phases. How to transmit the aggregation key from trusted server (corresponding to TTP in [29]) to FD (corresponding to PCS in [29]) is not demonstrated. If the secure channel is used, such cost is high. So, we consider that the consumption information is transmitted in public channels. At the same time, the whole scheme in [29] does not even reach CPA security, and the cost of time and communication increases in our scheme is rewarding.

Table 5: Time cost comparison (ms).

	PH1	PH2	PH3	PH4	PH5
Ours	$6T_s + 3T_h + 6T_a = 1.7591$	$(4T_s + 3T_h + 6T_a)\text{km} \le \text{m} = 1.1795\text{km}$	$3T_s + 6T_h + T_i = 0.9148$	$5T_s + 4T_h + 4(k-1)T_a$	$4T_s + 4T_h + 2T_m + 6T_a + 2T_{PR} = 124.1195$
[29]	$5T_s + T_{se} + 2T_h = 1.4688$	$(3T_s + T_{se} + T_h + 2T_a + 2T_{PR})\text{km} = 123.8264\text{km}$	$T_s + T_h + T_i + 2T_m = 0.3230$	$2(k+1)T_s + 4kT_a = 0.5869k + 0.5796$	$2T_m + 2T_a + 2T_{PR} = 124.1593$
[20]	$T_{2e} + T_e + T_h + T_m = 0.6602$			$(k+1)T_{2e} + 2T_e + (4k-1)T_m + 3kT_h = 0.3690k + 0.6537$	$T_{2e} + T_e + t_m + T_h + T_i + T_{PR} = 62.516$
[22]	$T_{2e} + T_e + T_h + T_m = 0.9644$			$2kT_{2e} + (k+1)T_e + (k+1)T_h + (2k-1)T_m = 1.0147k + 0.304$	$T_{2e} + T_e + T_h + 3T_m + T_{PR} = 62.1302$

Schemes in [20, 22] belong to the same type. They both lack the communication between the servers which calculates the aggregation data and the media device, like collector or gateway. In Ding et al.'s scheme [20], the message from the smart meter to the collector has 1024 + 1024 + 1024 + 1024 + 64 + 32 = 4192 bits and in total 4192 km bits. The message from the collector to the electricity service provider has the same construction as the last, and there are 4192 m bits. Finally, we could see that 4192 km + 4192 m bits occur in the whole process. Similarly, in Wang et al.'s scheme [22], the entire communication cost is 3616 km + 3552 m bits. However, both of them have weaknesses including lack of user anonymity and no consideration of traceability, which we have mentioned in Section 5. Also, no statistic data are deduced on service providers on both of them [20, 22].

Above all, our scheme is better than other schemes in [20, 22, 29] for security and practicality.

7. Conclusion

In this study, based on industrial Internet of Things, we give a novel scheme on smart grid, getting user power consumption and statistical data simultaneously. Formal proof with random oracle condition is shown to illustrate CPA security of the presented scheme. We also compare our scheme with some relative schemes for smart grid, and all can see ours is the only one that satisfies the security requirements. Via time and communication cost study, we express that our scheme performs well and it is fit for practicality.

The security level is an important index to evaluate the scheme. In the future, we will try to enhance the security level of such scheme, e.g., designing a new scheme that resists chosen-ciphertext attack and meets the practical requirements like tracking the concrete power consumption of every user and not only owns the function of aggregating data.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Fan Wu was supported by Xiamen University Tan Kah Kee College Scientific Research Foundation (JG2022SRF02). Professor Xiong Li was supported by the National Natural Science Foundation of China (62072078).

References

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

[2] F. Wu, L. Xu, X. Li, S. Kumari, M. Karuppiah, and M. S. Obaidat, "A lightweight and provably secure key agreement system for a smart grid with elliptic curve cryptography," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2830–2838, 2019.

[3] F. Wu, X. Li, L. Xu, S. Kumari, D. Lin, and J. J. P. C. Rodrigues, "An anonymous and identity-trackable data transmission scheme for smart grid under smart city notion," *Annals of Telecommunications*, vol. 75, no. 7-8, pp. 307–317, 2020.

[4] F. Wu, X. Li, L. Xu, and S. Kumari, "A privacy-preserving scheme with identity traceable property for smart grid," *Computer Communications*, vol. 157, pp. 38–44, 2020.

[5] Y. Gong, Y. Cai, Y. Guo, and Y. Fang, "A privacy-preserving scheme for incentive-based demand response in the smart grid," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1304–1313, 2016.

[6] M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. S. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 675–685, 2011.

[7] K. Mahmood, S. Ashraf Chaudhry, H. Naqvi, T. Shon, and H. Farooq Ahmad, "A lightweight message authentication scheme for smart grid communications in power sector," *Computers & Electrical Engineering*, vol. 52, pp. 114–124, 2016.

[8] J.-L. Tsai and N.-W. Lo, "Secure anonymous key distribution scheme for smart grid," *IEEE Transactions on Smart Grid*, vol. 7, pp. 906–914, 2016.

[9] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Transactions on Smart Grid*, vol. 9, p. 1, 2016.

[10] D. He, H. Wang, M. K. Khan, and L. Wang, "Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography," *IET Communications*, vol. 10, no. 14, pp. 1795–1802, 2016.

[11] X. Li, F. Wu, S. Kumari, L. Xu, A. K. Sangaiah, and K.-K. R. Choo, "A provably secure and anonymous message authentication scheme for smart grids," *Journal of Parallel and Distributed Computing*, vol. 132, pp. 242–249, 2019.

[12] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Generation Computer Systems*, vol. 81, pp. 557–565, 2018.

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

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

[15] F. Wu, X. Li, L. Xu, P. Vijayakumar, and N. Kumar, "A novel three-factor authentication protocol for wireless sensor networks with iot notion," *IEEE Systems Journal*, vol. 15, no. 1, pp. 1120–1129, 2021.

[16] F. Diao, F. Zhang, and X. Cheng, "A privacy-preserving smart metering scheme using linkable anonymous credential," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 461–467, 2015.

[17] H. Qu, P. Shang, X. J. Lin, and L. Sun, "Cryptanalysis of a privacy-preserving smart metering scheme using linkable anonymous credential," *IACR Cryptology ePrint Archive 2015*, vol. 1066, 2015.

[18] Z. Sui, M. Niedermeier, and H. de meer, "Tai: a threshold-based anonymous identification scheme for demand-response in smart grids," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3496–3506, 2018.

[19] S. Ge, P. Zeng, and K.-K. R. Choo, "An enhanced anonymous identification scheme for smart grids," in *Proceedings of the International Conference on Applications and Techniques in Cyber Security and Intelligence*, pp. 329–337, Fuyang, China, June2021.

[20] Y. Ding, B. Wang, Y. Wang, K. Zhang, and H. Wang, "Secure metering data aggregation with batch verification in industrial smart grid," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6607–6616, 2020.

[21] 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, pp. 15639–15648, 2021.

[22] J. Wang, L. Wu, S. Zeadally, M. K. Khan, and D. He, "Privacy-preserving data aggregation against malicious data mining attack for iot-enabled smart grid," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–25, 2021.

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

[24] D. He, S. Zeadally, H. Wang, and Q. Liu, "Lightweight Data Aggregation Scheme against Internal Attackers in Smart Grid Using Elliptic Curve Cryptography," *Wireless Communications and Mobile Computing 2017*, vol. 13, 2017.

[25] Z. Wang, "An identity-based data aggregation protocol for the smart grid," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2428–2435, 2017.

[26] 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.

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

[28] 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, 2019.

[29] H. Wang, D. He, and S. Zhang, "Balanced anonymity and traceability for outsourcing small-scale data linear aggregation in the smart grid," *IET Information Security*, vol. 11, no. 3, pp. 131–138, 2017.

[30] S. Wong, "Electricity consumption in china between 2010 and 2019," 2020, https://www.statista.com/statistics/302203/china-electricity-consumption/.

[31] D. J. Bernstein and T. Lange, "Computing small discrete logarithms faster," in *Proceedings of the International Conference on Cryptology in India*, pp. 317–338, Kolkata, India, December 2012.

[32] J. M. Pollard, "Monte Carlo methods for index computation (modp)," *Mathematics of Computation*, vol. 32, no. 143, pp. 918–924, 1978.

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

[34] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," in *Proceedings of the Annual International Cryptology Conference*, pp. 26–45, Springer, Santa Barbara, California, August 1998.

[35] Wikipedia, "Smart Meter," 2018, https://en.wikipedia.org/wiki/Smart_meter.

[36] B. Mitchell, "The Role of Fiber Optic Cables in Computer Networking," 2018, https://www.lifewire.com/fiber-optic-cable-817874.