Journal of Computers Vol. 35 No. 5, October 2024, pp. 73-90
doi: 10.53106/199115992024103505006

SERDA:
Secure Enhanced and Robust Data Aggregation Scheme for Smart Grid

Yiming Chen[1,4], Hongjun Duan[2], Dong Wang[3,6*],
Yining Liu[4], and Chin-Chen Chang[5]

[1] School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
yee_ming@foxmail.com

[2] Zhengzhou Art Infant Normal School, Zhengzhou 450000, China
13938566788@163.com

[3] School of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin 541004, China
wangdong918@guet.edu.cn

[4] School of Data Science and Artificial Intelligence, Wenzhou University of Technology, Wenzhou 325027, China
lyn7311@sina.com

[5] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan, ROC
ccc@fcu.edu.tw

[6] Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation Guilin 541004, China

Received 1 April 2024; Revised 12 April 2024; Accepted 13 April 2024

Abstract. Data aggregation is considered a viable security and privacy solution for smart grid as it allows to obtain the total electricity consumption within a region without disclosing individual data. However, existing data aggregation schemes give little consideration in their threat models to use cases where devices operate in untrustworthy environments and adversaries have physical system access, which is common in the smart grid. They cannot support authentication and resist physical attacks while maintaining data privacy and supporting fault tolerance for smart meter (SM) failures. Motivated by this, a secure enhanced and robust data aggregation (SERDA) scheme for smart grid is introduced in this article. The SERDA scheme provides enhanced security for key storage and updates based on physically unclonable function (PUF), while supporting data privacy protection and fault tolerance for SM failures without reliance on a trusted third party (TTP). Security analysis and performance evaluation demonstrate that SERDA meets the excepted goals and is efficient compared with related work.

Keywords: authentication, data aggregation, enhanced security, robust, smart grid

1 Introduction

The contemporary power grid is undergoing a transformative shift towards the smart grid. Within such a cyber-physical system, each node possesses computation and communication capabilities, which empower these nodes to locally process data and engage in data exchange [1, 2]. For example, it allows the encryption of meter readings on the smart meter (SM) and supports bidirectional communication between utility companies and their customers [2], enabling utility companies to efficiently aggregate individual electricity consumption data within certain regions, monitor the power grid in real-time, and balance power loads.

Despite the advantages of the smart grid, its widely deployed infrastructure also faces a variety of attacks, which has raised widespread concerns about the security and privacy of electricity consumption data. Due to the

* Corresponding Author

fact that smart meters and other devices are usually deployed in outdoor open environments and have limited self-protection capabilities, they are easily targeted by attackers [3]. Attackers may intercept communication between electricity meters and control centers through physical intrusion or network attacks, steal actual electricity consumption data such as meter readings, or directly invade terminal devices to steal or tamper with confidential data stored therein. These attacks will interfere with power distribution scheduling, and jeopardize the stability of the grid, even lead to substantial financial losses [4]. Additionally, the detailed electricity consumption data can reveal sensitive customer information, such as individuals' habits and financial status, raising privacy concerns or even violating local data regulations. Therefore, addressing these security and privacy issues is crucial from both an industry and an individual perspective.

Data aggregation schemes are introduced as preferred solutions to deal with the above issues, such as those proposed in [5, 6] and [7]. The general process for data aggregation schemes is as follows: SMs encrypt the individual consumption data, then generate corresponding reports and transmit them to the aggregator (AG). AG aggregates the reports and conveys the aggregated report to the control center (CC) for decryption. However, a vulnerability arises in this process from the collusion attack between internal attackers CC and AG, which are typically managed by the same utility company in practice. Since AG has access to individual reports, CC can also access the report and obtain individual consumption data in collusion with AG.

In response to potential threats from internal attackers, mask-based data aggregation has been introduced, as discussed in [8]. This approach involves a trusted third party (TTP) responsible for assigning blinding factors with a total of zero to mask individual electricity consumption data. By adopting this strategy, the blinding factors do not affect the final aggregated report, and CC cannot access individual consumption data since it has no knowledge about the factors assigned by TTP. Nevertheless, establishing the complete trustworthiness of an entity in practice poses a challenge. For example, the employee of the entity may compromise individuals' data privacy for personal gain. Besides, this approach requires SMs to stay online. Otherwise, the blinding factors cannot be removed, leading to the wrong aggregation result. Consequently, pairwise blinding is introduced as an effective approach, which has gained widespread recognition through its adoption in [9] for dealing with client failures. This approach requires pairs of SMs to use the Diffie-Hellman key exchange to agree on pairwise blinding factors with a sum of zero to mask their readings while eliminating the need for TTP to allocate blinding factors. Therefore, CC remains oblivious to the pairwise blinding factors since it doesn't participate in the negotiation of these factors, as demonstrated in [10]. Meanwhile, due to its ability to handle client failures, it can also achieve the requirement of fault tolerance, enhancing the robustness of the scheme.

However, the above method must ensure that the pairwise blinding factors are securely negotiated and stored. Specifically, SMs should authenticate each other's identities to securely negotiate these factors. In this manner, they are also required to securely store secrets, such as the secret key for identity authentication. Moreover, devices such as SMs are usually deployed in unmanned and open locations with limited resources and poor self-protection capabilities, which makes them prone to physical attacks, as mentioned before. In this case, an attacker can use physical attacks to extract secrets from SMs or create a clone one, compromising the protocol's security, such as through an impersonation attack. Actually, existing data aggregation schemes cannot support authentication and resist physical attacks while maintaining data privacy and supporting fault tolerance for SM failures. They typically assume that SMs are trusted or honest-but-curious entities and do not consider the use case of SMs facing physical attacks in their threat models.

To deal with such issues, a secure enhanced and robust data aggregation (SERDA) scheme for smart grid is proposed in this article. The proposed scheme provides enhanced security for key storage and updates based on PUF, while supporting data privacy protection and fault tolerance for SM failures without reliance on TTP. Our contributions can be summarized as follows:

1) To provide enhanced security for key storage, one-time pads (OTPs) are generated using PUF and fuzzy extractor (FE) to mask the stored keys in devices. The OTP can achieve perfect secrecy when the length of the OTP is at least equal to the length of the plaintext. Furthermore, since the OTP is generated by PUF and FE, and the PUF possesses resistance to physical attacks, an adversary cannot exploit the PUF responses through physical attacks to regenerate the OTP and decrypt the stored keys, thereby enhancing physical security.
2) To provide enhanced security for key updates, an authenticated key agreement protocol based on PUF is designed to provide mutual authentication and implicit physical security authentication for blinding factor updates between SMs. Mutual authentication ensures that the blind factors are updated securely between SMs. While implicit physical security authentication ensures that SMs are not physically attacked during the update process.

3) Besides, data privacy protection and fault tolerance for SM failures without reliance on TTP are also achieved based on the designed authenticated key agreement protocol. SERDA also supports batch verification to improve verification efficiency.
4) To demonstrate the security and efficiency of SERDA, a detailed security analysis of SERDA is conducted and the performance evaluation is made between SERDA and recent related work.

The article is organized as follows: Section 2 presents the related work. In Sections 3 and 4, the preliminaries and the system models are introduced, respectively. Section 5 provides a detailed presentation of SERDA, followed by the security analysis of SERDA in Section 6. Section 7 compares SERDA's performance to that of related work, and Section 8 concludes this article.

2 Related Work

In this section, a detailed overview of relevant data aggregation schemes is provided, along with a functional comparison of the discussed schemes given in Table 1.

Considering the huge communication and computation costs brought by the high-frequency aggregation of multidimensional data with numerous customers, Lu et al. [7] proposed the EPPA scheme. In this scheme, the super-increasing sequence is employed to realize the functionality of expressing multidimensional data in a single-dimensional form. In smart grid, the deployment of public key infrastructure (PKI) incurs huge costs due to the large user base. Therefore, Wang et al. [11] utilize the additive homomorphic property of identity-based cryptography to aggregate the data while verifying the identity without the need to maintain a PKI in the scheme they proposed. Besides, access control also attracts attention. Lang et al. [12] leverage attribute-based encryption (ABE) to manage the private keys of each dimensional data in their proposed multidimensional data aggregation scheme. Therefore, access to each dimension of multidimensional data can be finely controlled. In order to achieve efficient authentication, Shang et al. [13] utilizes the ECDSA signature in their proposed data aggregation scheme. However, these schemes require AG to be trustworthy. Otherwise, CC and AG can launch collusion attacks to access individual consumption data, which is also known as the internal attack in [8]. Similarly, the above issues are present in [14] and [15].

Table 1. Comparison of functional features[1]

Scheme	Year	Auth	BV	PS	FT	NTTP	CR
[7]	2012	√	√	×	×	×	√
[8]	2014	√	√	×	×	×	√
[10]	2021	×	×	×	√	√	√
[11]	2017	√	√	×	√	√	×
[12]	2018	×	×	×	√	×	×
[13]	2023	√	√	×	√	√	×
[14]	2020	√	√	×	√	√	×
[15]	2020	√	√	×	√	√	×
[16]	2023	√	√	×	×	×	√
[17]	2023	×	×	×	×	×	√
[18]	2022	√	√	×	×	×	√
[19]	2020	√	×	×	√	×	×
[20]	2021	√	√	×	×	×	√
[21]	2020	√	√	×	×	×	√
[22]	2021	√	√	×	×	√	√
[23]	2019	√	√	×	×	√	√
[24]	2021	√	√	×	×	√	√
[25]	2020	×	×	×	√	√	√
[26]	2018	×	×	×	√	√	√
[27]	2023	√	√	×	√	√	√
SERDA	-	√	√	√	√	√	√

1 √ means the functionality is provided, × means the functionality is not provided. Auth, BV, PS, FT, NTTP, CR denote Authentication, Batch Verification, Physical Security, Fault Tolarance, No need for TTP, and Collusion Resistance, respectively.

Fan et al. [8] noticed the threat from internal attackers and devised a method in their proposed scheme to avoid internal attacks by injecting blinding factors assigned by TTP into the data. However, there are some drawbacks due to the reliance on computing the discrete logarithm problem (DLP): CC needs to compute the DLP to decrypt the aggregated ciphertext, which is computationally inefficient; the space for plaintext (i.e., the meter reading) is limited. Moreover, it cannot deal with SM failures. Despite the drawbacks, this approach of utilizing a TTP for assigning blinding factors is widely used in [16-22]. Wu et al. [16] considered malicious aggregators that may return wrong results and proposed the LVSA-MD scheme. Homomorphic MAC and identity-based signature are used to achieve result verification and source authentication, respectively. Zhang et al. [17] proposed the FSDA scheme, in which individual consumption data is divided into multiple slots, allowing CC to adjust the subset size flexibly. Maintaining data integrity in multi-type data aggregation has also received attention. Zhang et al. [18] proposed the VPMDA scheme, which utilizes the BLS signature to support integrity verification. In spite of the strengths of the above schemes, they all rely on TTP. Such trusted assumptions are hardly guaranteed in practice.

In 2018, Liu et al. [23] proposed the 3PDA scheme for smart grid. The authors utilize the distributed decryption technology to achieve the goal of eliminating reliance on TTP. A similar approach was taken in [24]. However, there are some limitations. The distributed decryption technology requires SMs to remain online, which lacks robustness since SM failures are common in the grid. To address this concern, some schemes have aimed to eliminate dependence on a TTP while maintaining robustness to meter faults [10, 25-27]. In [10], the blinding factors are updated using pairwise blinding factors negotiated through Diffie-Hellman key exchange. Fault tolerance is achieved by removing pairwise blinding factors from faulty SMs. While Xue et al. [25] utilized secret sharing to recover blinding factors of fault SMs in their proposed scheme. Knirsch et al. [26] proposed a new approach for utilizing blinding factors through user obfuscation. The first SM adds its own blinding factor to the initial blinding factor from CC to get its sum for encryption and passes the sum to the next SM, which takes a similar action. Finally, CC obtains the sum of all factors from the last SM. Considering customers' willingness to share data, Zeng et al. [27] designed a data encryption mechanism that includes two encryption methods, allowing SMs to report with blinding factors masking or not. All of these schemes achieve the goal of eliminating dependence on TTP while maintaining robustness. However, due to the need for communication between SMs, mutual authentication between SMs needs to be considered, which is rarely discussed.

In addition, SMs also face the risk of physical attacks, which make data aggregation schemes fragile since the malicious SMs that have been physically attacked are difficult to detect while SMs are typically assumed to be honest. Therefore, SERDA is proposed to provide authentication and physical attack resistance while maintaining data privacy and fault tolerance for SM failures without relying on TTP.

3 Preliminaries

In this section, we introduce some basic knowledge of the proposed SERDA scheme.

3.1 Elliptic Curve Cryptography

Let G be an elliptic curve group with prime order q and generator P in elliptic curve cryptography (ECC) [28]. Two operations are defined on the group G. The first one, point addition, is defined geometrically on this group, which is usually denoted by the symbol "+". While the second one, scalar multiplication, is defined as repeated point addition, denoted by the symbol "$\cdot$". For example, for $P \in G$ and $a \in Z_q^*$, $aP = a \cdot P = P + P + \cdots P$ (a times). Based on the group \$G\$, two computational hardness problems are given as follows:

Discrete Logarithm Problem (DLP): Given a tuple (P, aP) where $P \in G$ and $a \in Z_q^*$, the DLP is said to be hard when it is computational infeasible to compute a in probabilistic polynomial time (PPT).

Computational Diffie-Hellman Problem (CDHP): Given a tuple (P, aP, bP) where $P \in G$ and $a, b \in Z_q^*$, the CDHP is said to be hard when it is computational infeasible to compute abP in PPT.

3.2 Physically Unclonable Function

PUF utilizes the unclonable manufacturing variability of the inherent, unique structure of the semiconductor device to generate unique responses for specific challenges [29]. Formally, given a random challenge C of x-bit, PUF outputs a unique response $R = PUF(C)$ of y-bit. PUF can resist physical attacks based on the following properties.

1) Given a challenge C, computing its unique response R is easy.
2) Given a challenge C, its unique response R is randomly generated based on the physical characteristics of the PUF. An attacker cannot predict the response to a new, randomly selected challenge from a polynomial-sized sample of adaptively chosen challenge-response pairs (CRPs).
3) Manufacturing two PUFs with the same response is not feasible.

3.3 Fuzzy Extractor (FE)

FE is a cryptographic primitive that provides the same output for similar inputs within a certain range of noise [30]. In SERDA, it is employed to avoid the interference of subtle environmental noise on PUF.

Formally, FE generates a key K and a help string hs: $(K, hs) = FE.Gen(R)$ through the key generation algorithm $FE.Gen()$ for a given random string input R. And for input within the tolerable noise range, FE is able to recover $K = FE.Rec(R', hs)$ through the key recovery algorithm $FE.Rec()$ with the help string hs and the input R' with noise.

4 System Model

4.1 Communication Model

As depicted in Fig. 1, three entities are involved in the scheme, namely, SMs, AG, and CC.

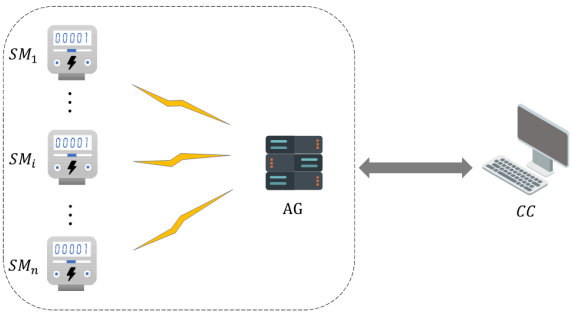


Fig. 1. Communication model

SMs: SMs are installed on the user side to report the customer's data and send the encrypted report data to AG for aggregation. Moreover, PUF and FE are integrated into SMs to provide physical security.

AG: AG is protected by the integrated PUF and FE as well and is responsible for checking if the reports are valid. Then, it aggregates the report and sends it to CC for decryption.

CC: In SERDA, all relevant system parameters are generated and published by CC. Also, CC is required to complete registration for SM and AG and decrypt the aggregated report data from AG.

4.2 Threat Model

In SERDA, both internal and external threats are considered. Under the honest-but-curious assumption, the proposed SERDA scheme will be strictly adhered to by the internal entities CC, AG, and SMs. However, they may be interested in others' information and try to obtain it by analyzing the received messages. Moreover, CC and AG may be managed by the same utility company, so they might collude to obtain sensitive information regarding SMs by sharing accessible data. Also, it is assumed that an external adversary $\mathcal{A}$ is considered as follows:

1) $\mathcal{A}$ has the ability to eavesdrop on the communication channels from SM to AG and from AG to CC, and may try to intercept the encrypted report data and the aggregated report data.
2) $\mathcal{A}$ may impersonate the identity of the legitimate smart meter.
3) $\mathcal{A}$ may launch physical attacks to extract secret keys stored in SM and AG, but it is difficult to crack PUF.

4.3 Design Goals

The design goal of SERDA is to provide authentication and physical security for devices, including SMs and AG, while maintaining the privacy of individual electricity consumption data to prevent it from being disclosed. The details are as follows:

1) **Physically secure**: Considering the threat of physical attacks on SMs and AG, the keys stored in the devices need to be protected.
2) **Mutual authentication**: When two smart meters SM_i and SM_j collaborate to update their blinding factors, they should authenticate each other mutually.
3) **Privacy-preservation**: The privacy of individual electricity consumption data should be protected, i.e., no one other than the customer himself could know individual electricity consumption data.

5 The Proposed Scheme

This section presents SERDA in detail, with some of the notations presented in Table 2 and the workflow of SERDA shown in Fig. 2.

Table 2. Definitions of notations

Notations	Definitions
λ	A security parameter
G, q, P	An elliptic curve group G with prime order q and generator P
PK_C, s	CC's public-private key pair
k_{sum}, k_i	The sum of the blinding factors and SM_i's blinding factor
H	A secure hash function
C_i, R_i	The CRPs of PUF integrated in SM_i
K_i, hs_i	The secret key and help string generated by FE integrated in SM_i
B_{ag}, s_{ag}	AG's public-private key pair
$(A_i, B_i), (a_i, s_i)$	SM_i's public-private key pair
Sp_{ag}	AG's protected private key
Ap_i, Sp_i, Kp_i	SM_i's protected secret keys
$k_{i,j}$	The pairwise factors negotiated by SM_i and SM_j

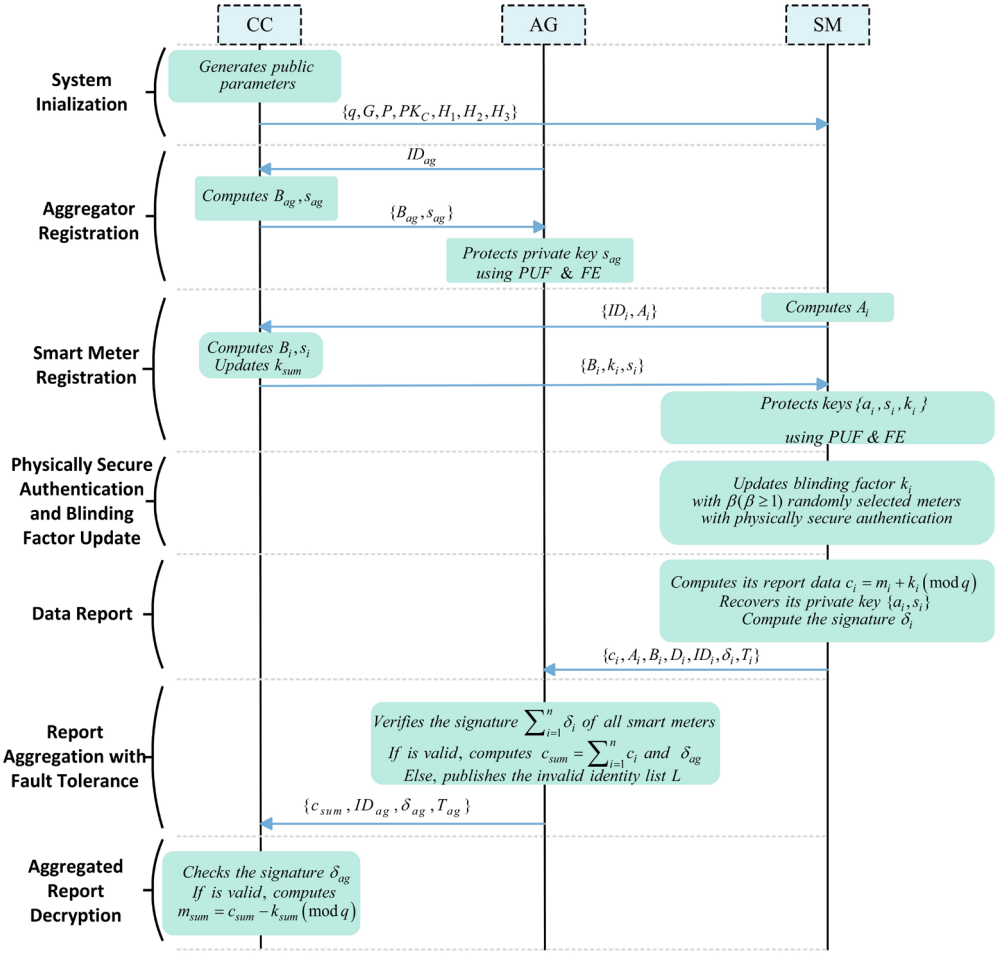


Fig. 2. Workflow of SERDA

5.1 System Initialization

In this phase, CC generates the system parameters for running the proposed SERDA scheme based on the given security parameter λ and publishes the system parameters to AG and SMs.

1) CC generates the relevant parameters (G, q, P) of the elliptic curve group based on the given security parameter λ, where G is the elliptic curve group with prime order q and generator P.
2) Then, $s \xleftarrow{\$} Z_q^*$ is chosen as CC's private key, and $PK_C = s \cdot P$ is set as the corresponding public key. And then, CC sets $k_{sum} = 0$ and selects a secure hash function $H: \{0,1\}^* \to Z_q^*$.
3) CC publishes $\{q, P, G, PK_C, H\}$.

5.2 Aggregator Registration

In this phase, the aggregator AG should register as a legitimate participant with CC when it is deployed.

1) AG sends its identity ID_{ag} to CC for registration via a secure channel.
2) Upon receiving ID_{ag} from AG, CC checks whether the request is valid. If not, abort. Otherwise, CC selects $b_{ag} \xleftarrow{\$} Z_q^*$ and computes $B_{ag} = b_{ag} \cdot P$, $s_{ag} = b_{ag} + s \cdot H(ID_{ag} \| B_{ag})$. And then, CC returns $\{B_{ag}, s_{ag}\}$ to AG via a secure channel.
3) Upon receiving $\{B_{ag}, s_{ag}\}$, AG generates a challenge C_{ag} and gets the response $R_{ag} = PUF(C_{ag})$. Then, it generates $(K_{ag}, hs_{ag}) = FE.Gen(R_{ag})$ through the fuzzy extractor, and computes the OTP $tem_{ag} = H(K_{ag})$ and masks the key as $Sp_{ag} = tem_{ag} \oplus s_{ag}$. Here, the hash function H further boosts the entropy of K_{ag}.
4) Finally, AG stores $\{Sp_{ag}, B_{ag}, ID_{ag}, C_{ag}, hs_{ag}\}$.

5.3 Smart Meter Registration

SM_i should firstly register as a legitimate participant with CC when it is installed at the customer side.

1) SM_i with identity ID_i selects $a_i \xleftarrow{R} Z_q^*$ to compute $A_i = a_i \cdot P$.
2) Then, SM_i sends $\{ID_i, A_i\}$ to CC for registration via a secure channel.
3) Upon receiving $\{ID_i, A_i\}$, CC selects $b_i \xleftarrow{R} Z_q^*$ and computes: $B_i = b_i \cdot P$, $s_i = s \cdot H(A_i \| B_i \| ID_i) + b_i$.
4) CC selects $k_i \xleftarrow{R} Z_q^*$ and sends $\{B_i, k_i, s_i\}$ to SM_i via a secure channel and updates $k_{sum} = k_{sum} + k_i$.
5) Upon receiving $\{B_i, k_i, s_i\}, \{a_i, s_i\}$ is set as the private key of SM_i, and correspondingly, is set as its public key.
6) Then, SM_i generates a challenge C_i to compute $R_i = PUF(C_i)$, $(K_i, hs_i) = FE.Gen(R_i)$. The corresponding OTPs are computed as $x_i = H(K_i \| 0)$, $y_i = H(K_i \| 1)$. And the secret keys are masked as $Ap_i = x_i \oplus a_i$, $Sp_i = y_i \oplus s_i$, $Kp_i = H(x_i \| y_i) \oplus k_i$.
7) Finally, SM_i stores $\{Ap_i, Sp_i, Kp_i, ID_i, C_i, hs_i\}$.

5.4 Physically Secure Authentication and Blinding Factor Update

After the newly joined smart meter SM_i takes part in the data aggregation area, it needs to randomly select $\beta(\beta \geq 1)$ meters in the data aggregation area to update blinding factors collaboratively. Assuming that $SM_j(j < i)$ is one of the β meters, SM_i and SM_j collaborate to update their blinding factors through the following authenticated key agreement protocol:

1) SM_i generates a timestamp T_1, a random challenge $C_{i,j}$, selects $e_i \xleftarrow{R} Z_q^*$, and computes: $R_{i,j} = PUF(C_{i,j})$, $(K_{i,j}, hs_{i,j}) = FE.Gen(R_{i,j})$, $E_i = H(e_i \| K_{i,j}) \cdot P$.
2) Then, SM_i sends $\{ID_i, E_i, T_1\}$ to SM_j.
3) Upon receiving $\{ID_i, E_i, T_1\}$, SM_j checks if T_1 is fresh. If not, abort. Otherwise, it generates a challenge $C_{j,i}$, a timestamp T_2, selects $e_j \xleftarrow{R} Z_q^*$, and computes: $R_{j,i} = PUF(C_{j,i})$, $(K_{j,i}, hs_{j,i}) = FE.Gen(R_{j,i})$, $E_j = H(e_j \| K_{j,i}) \cdot P$.
4) Then, SM_j recovers its secret keys: $R_{j'} = PUF(C_j)$, $(K_j) = FE.Rec(R_{j'}, hs_j)$, $x_j = H(K_j \| 0)$,

$y_j = H(K_j \| 1)$, $k_j = Kp_j \oplus H(x_j \| y_j)$, $a_j = x_j \oplus Ap_j$, $s_j = y_j \oplus Sp_j$.

5) Thereafter, SM_j computes: $\omega_j = s_j \cdot E_i$, $v_j = H(ID_j \| A_j \| B_j \| \omega_j \| T_2)$, $\sigma_j = s_j + a_j + v_j \cdot H(e_j \| K_{j,i})$, and sends $\{ID_j, A_j, B_j, E_j, \sigma_j, T_2\}$ to SM_i.

6) Upon receiving $\{ID_j, A_j, B_j, E_j, \sigma_j, T_2\}$, SM_i checks if T_2 is fresh. If not, abort. Otherwise, it computes: $R_{i,j'} = PUF(C_{i,j})$, $(K'_{i,j}) = FE.Rec(R'_{i,j}, hs_{i,j})$, $\omega_{j'} = H(e_i \| K'_{i,j}) \cdot (H(A_j \| B_j \| ID_j) \cdot PK_C + B_j)$, $v_{j'} = H(ID_j \| A_j \| B_j \| \omega_{j'} \| T_2)$.

7) Then, SM_i checks whether the equation holds.

$$\sigma_j \cdot P = H(A_j \| B_j \| ID_j) \cdot PK_C + B_j + A_j + v_{j'} \cdot E_j \quad (1)$$

If not, abort. Otherwise, SM_i authenticates SM_j. Then, it recovers its secret keys by computing: $R_i = PUF(C_i)$, $(K_i) = FE.Rec(R_i, hs_i)$, $x_i = H(K_i \| 0)$, $y_i = H(K_i \| 1)$, $k_i = Kp_i \oplus H(x_i \| y_i)$, $a_i = x_i \oplus Ap_i$, $s_i = y_i \oplus Sp_i$.

8) Thereafter, SM_i generates a timestamp T_3, computes: $\omega_i = s_i \cdot E_j$, $k_{i,j} = H(\omega_i \| ID_i \| ID_j)$, $v_i = H(k_{i,j} \| ID_i \| ID_j \| A_i \| B_i \| T_3)$, $\sigma_i = s_i + a_i + v_i \cdot H(e_i \| K_{i,j})$, and sends $\{A_i, B_i, \sigma_i, T_3\}$ to SM_j.

9) Upon receiving $\{A_i, B_i, \sigma_i, T_3\}$, SM_j checks whether T_3 is fresh. If not, abort. Otherwise, it computes: $R'_{j,i} = PUF(C_{j,i})$, $(K'_{j,i}) = FE.Gen(R'_{j,i}, hs_{j,i})$, $\omega_{i'} = H(e_j \| K'_{j,i}) \cdot (H(A_i \| B_i \| ID_i) \cdot PK_C + B_i)$, $k'_{i,j} = H(\omega_{i'} \| ID_i \| ID_j)$, $v_{i'} = H(k'_{i,j} \| ID_i \| ID_j \| A_i \| B_i \| T_3)$.

10) Then, SM_j checks whether the equation holds.

$$\sigma_i \cdot P = H(A_i \| B_i \| ID_i) \cdot PK_C + B_i + A_i + v_{i'} \cdot E_i \quad (2)$$

If not, abort. Otherwise, SM_j authenticates SM_i. At this point, SM_i and SM_j successfully complete the physical secure mutual authentication and share the same pairwise factor $k_{i,j}$. Thereafter, SM_i updates its blinding factor as $k_i = k_i + k_{i,j}$ and stores $Ek_i = k_{i,j} \oplus H(a_i)$. SM_j updates its blinding factor as $k_j = k_j - k_{i,j}$ and stores $Ek_j = k_{i,j} \oplus H(a_j)$.

5.5 Data Report

In this phase, SM_i executes the following step to report its data to AG.

1) SM_i recovers its secret keys by computing:
2) $R_i = PUF(C_i)$, $(K_i) = FE.Rec(R_{i'}, hs_i)$, $x_i = H(K_i \| 0)$, $y_i = H(K_i \| 0)$, $k_i = Kp_i \oplus H(x_i \| y_i)$, $a_i = x_i \oplus Ap_i$, $s_i = y_i \oplus Sp_i$.
3) SM_i encrypts its report data m_i by computing: $c_i = m_i + k_i \pmod q$.
4) SM_i generates a timestamp T_i and selects $d_i \xleftarrow{R} Z_q^*$ to compute:
5) $\delta_i = s_i + a_i + d_i \cdot H(c_i \| ID_i \| ID_{ag} \| T_i)$, $D_i = d_i \cdot P$.
6) SM_i sends $\{c_i, A_i, B_i, D_i, ID_i, \delta_i, T_i\}$ to AG.

5.6 Report Aggregation with Fault Tolerance

Suppose there are n SMs in the aggregation area. Upon receiving n reports, AG verifies the reports and sends the aggregated ciphertext to CC.

1) AG first checks whether T_i, $i \in [1,n]$ is fresh. If all the timestamps are fresh. AG checks whether the equation (3) holds.

$$(\sum_{i=1}^{n} \delta_i) \cdot P = (\sum_{i=1}^{n} H(A_i \| B_i \| ID_i)) \cdot PK_C + \sum_{i=1}^{n} (B_i) + \sum_{i=1}^{n} (A_i) + \sum_{i=1}^{n} (H(c_i \| ID_i \| ID_{ag} \| T_r)) \cdot D_i \quad (3)$$

2) If the equation holds, AG aggregates the ciphertexts $c_{sum} = \sum_{i=1}^{n} (m_i + k_i) \left(\mathrm{mod}\, q\right)$ and recovers its private key by computing: $R_{ag} = PUF(C_{ag})$, $K_{ag} = FE.Rec(R_{ag}, hs_{ag})$, $tem_{ag} = H(K_{ag})$, $s_{ag} = Sp_{ag} \oplus tem_{ag}$.

3) Then, AG generates a timestamp T_{ag} and selects $d_{ag} \overset{R}{\leftarrow} Z_q^*$ to compute:
$D_{ag} = d_{ag} \cdot P$, $\delta_{ag} = s_{ag} + d_{ag} \cdot H(c_{sum} \| ID_{ag} \| T_{ag})$, and then sends $\{c_{sum}, ID_{ag}, D_{ag}, \delta_{ag}, T_{ag}\}$ to CC.

4) If the equation does not hold or AG does not collect enough reports, it will check for the invalid SMs and publishes the corresponding identity list: $L = \{ID_{1'}, ID_{2'}, \cdots, ID_{k'}\}$. Then, each smart meter SM_i except for the invalid SMs in the aggregation area recovers the blinding factors and removes the corresponding pairwise factor $k_{i,j}$ from its blinding factor k_i as $k_i = k_i - k_{i,j} (i > j)$ or $k_i = k_i + k_{i,j} (i < j)$ for each $ID_j \in L$ and uploads its report data again for aggregation.

5.7 Aggregated Report Decryption

CC checks if T_{ag} is fresh, and verifies whether the equation (1) holds.

$$\delta_{ag} \cdot P = B_{ag} + H(ID_{ag} \| B_{ag}) \cdot PK_C + H(c_{sum} \| ID_{ag} \| T_{ag}) \cdot D_{ag} \quad (4)$$

If the check is passed, CC calculates $m_{sum} = c_{sum} - k_{sum} \left(\mathrm{mod}\, q\right)$, where m_{sum} is the total electricity consumption of the aggregation area.

6 Security Analysis

In this section, the security of SERDA is proved from three aspects, namely correctness analysis, formal security analysis, and informal security analysis.

6.1 Correctness Analysis

Theorem 1. If SMs, AG and CC correctly execute SERDA, then CC can obtain the correct aggregation result.

Proof. The proof of correctness of SERDA is depicted as follows.

SM_i and SM_j can correctly update their blinding factors.

$$\begin{aligned} \omega_j &= s_j \cdot E_i \\ &= (s \cdot H(A_j \| B_j \| ID_j) + b_j) \cdot (H(e_i \| K_{i,j'}) \cdot P) \\ &= H(e_i \| K_{i,j'}) \cdot (H(A_j \| B_j \| ID_j) \cdot PK_C + B_j) \\ &= \omega_{j'}. \end{aligned} \quad (5)$$

$$\begin{aligned}\sigma_j \cdot P &= (s_j + a_j + v_j \cdot H(e_j \| K_{j,i})) \cdot P \\ &= (s \cdot H(A_j \| B_j \| ID_j) + b_j + a_j + v_j \cdot H(e_j \| K_{j,i})) \cdot P \\ &= H(A_j \| B_j \| ID_j) \cdot PK_C + B_j + A_j + v_{j'} \cdot E_j.\end{aligned} \tag{6}$$

$$\begin{aligned}\omega_i &= s_i \cdot E_j \\ &= (s \cdot H(A_i \| B_i \| ID_i) + b_i) \cdot (H(e_j \| K_{j,i'}) \cdot P) \\ &= H(e_j \| K_{j,i'}) \cdot (H(A_i \| B_i \| ID_i) \cdot PK_C + B_i) \\ &= \omega_{i'}.\end{aligned} \tag{7}$$

$$\begin{aligned}k_{i,j} &= H(w_i \| ID_i \| ID_j) \\ &= H(w_{i'} \| ID_i \| ID_j) \\ &= k'_{i,j}.\end{aligned} \tag{8}$$

$$\begin{aligned}\sigma_i \cdot P &= (s_i + a_i + v_i \cdot H(e_i \| K_{i,j})) \cdot P \\ &= (s \cdot H(A_i \| B_i \| ID_i) + b_i + a_i + v_i \cdot H(e_i \| K_{i,j})) \cdot P \\ &= H(A_i \| B_i \| ID_i) \cdot PK_C + B_i + A_i + v_{i'} \cdot E_i.\end{aligned} \tag{9}$$

AG can correctly verify the reported data for aggregation.

$$\begin{aligned}(\sum_{i=1}^{n} \delta_i) \cdot P &= (\sum_{i=1}^{n} (s_i + a_i + d_i \cdot H(c_i \| ID_i \| ID_{ag} \| T_i))) \cdot P \\ &= (\sum_{i=1}^{n} H(A_i \| B_i \| ID_i)) \cdot PK_C + \sum_{i=1}^{n} (B_i) + \sum_{i=1}^{n} (A_i) + \sum_{i=1}^{n} (H(c_i \| ID_i \| ID_{ag} \| T_r) \cdot D_i)\end{aligned} \tag{10}$$

CC can correctly verify the aggregated reported data to obtain the result.

$$\begin{aligned}\delta_{ag} \cdot P &= (s_{ag} + d_{ag} \cdot H(c_{sum} \| ID_{ag} \| T_{ag})) \cdot P \\ &= (b_{ag} + s \cdot H(ID_{ag} \| B_{ag}) + d_{ag} \cdot H(c_{sum} \| ID_{ag} \| T_{ag})) \cdot P \\ &= B_{ag} + H(ID_{ag} \| B_{ag}) \cdot PK_C + H(c_{sum} \| ID_{ag} \| T_{ag}) \cdot D_{ag}.\end{aligned} \tag{11}$$

6.2 Formal Security Analysis

In this subsection, we analyze the authenticity of the phase **Physical Secure Authentication and Blinding Factor Update** of SERDA through the BAN logic, which is a widely used formula security verification tool. For better reading, let A and B be the entities involved, M and N be two messages, and K be an encryption key. Several modal operators are given below.

1) $A \mid\equiv M$: A acts as if M is true, and may assert M in other messages.
2) $A \lhd M$: A message containing M has been send to A by someone so that A can read and repeat M.
3) $A \mid\Rightarrow M$: A has jurisdiction over M.
4) $A \mid\sim M$: A sent a message containing M at some time.
5) (M, N) : M and N combine to form a message (M, N).
6) $\#(M)$: M is a fresh message that has not been sent before.
7) $A \overset{K}{\rightarrow} B$: A and B can communicate using the key K that is unknown to others.
8) $\overset{K}{\mapsto} A$: K is a published public key of A, and K^{-1} is the corresponding private key of A.

9) $\{M\}_K$: M is encrypted under the key K.

10) $A \overset{M}{\leftrightarrow} B$: M is a secret known only to A and B.

Then, some rules of inference are listed below based on the above modal operators.

1) R1. Message-meaning rule: $\dfrac{A \mid\equiv (\overset{K}{\mapsto} B), A \lhd \{M\}_{K^{-1}}}{A \mid\equiv (B \mid\sim M)}$.

2) R2. Nonce verification rule: $\dfrac{A \mid\equiv (\#(M)), A \mid\equiv (B \mid\sim M)}{A \mid\equiv (B \mid\equiv M)}$.

3) R3. Jurisdiction rule: $\dfrac{A \mid\equiv (B \mid\Rightarrow M), A \mid\equiv (B \mid\equiv M)}{A \mid\equiv M}$.

4) R4. Freshness rule: $\dfrac{A \mid\equiv (\#(M))}{A \mid\equiv (\#(M, N))}$.

5) R5. Session key rule: $\dfrac{A \mid\equiv (\#(K)), A \mid\equiv (B \mid\equiv M)}{A \mid\equiv (A \overset{K}{\leftrightarrow} B)}$.

6) R6. Belief rule: $\dfrac{A \mid\equiv (M, N)}{A \mid\equiv M}$.

The following goals needs to be achieved.

1) Goal 1: $SM_i \mid\equiv (SM_i \overset{k_{i,j}}{\leftrightarrow} SM_j)$.

2) Goal 2: $SM_j \mid\equiv (SM_j \overset{k_{i,j}}{\leftrightarrow} SM_i)$.

Then, we give the idealized form of the messages exchanged among SM_i and SM_j as below.

1) Message 1: $SM_i \rightarrow SM_j : \{ID_i, E_i, T_1\}$.

2) Message 2: $SM_j \rightarrow SM_i : \{ID_j, A_j, B_j, E_j, \sigma_j : \{ID_j, A_j, B_j, T_2, E_j\}_{\{a_j, s_j\}}, T_2\}$.

3) Message 3: $SM_i \rightarrow SM_j : \{A_i, B_i, \sigma_i : \{k_{i,j}, ID_i, ID_j, A_i, B_i, T_3, E_i\}_{\{a_i, s_i\}}, T_3\}$.

According to the description of the scheme, the following assumptions about the initial state are made which are listed as below.

1) $\mathbb{A}_1$: $SM_j \mid\equiv (\#(T_1))$.

2) $\mathbb{A}_2$: $SM_i \mid\equiv (\#(T_2))$.

3) $\mathbb{A}_3$: $SM_j \mid\equiv (\#(T_3))$.

4) $\mathbb{A}_4$: $SM_i \mid\equiv (\overset{A_j, B_j}{\mapsto} SM_j)$.

5) $\mathbb{A}_5$: $SM_i \mid\equiv (SM_j \mid\Rightarrow E_j)$.

6) $\mathbb{A}_6$: $SM_j \mid\equiv (\overset{A_i, B_i}{\mapsto} SM_i)$.

7) $\mathbb{A}_7$: $SM_j \mid\equiv (SM_i \mid\Rightarrow k_{i,j})$.

Subsequently, the aforementioned idealized form is analyzed.

According to Message 1, the following statement is obtained.

$S_1 : SM_j \mid\lhd \{ID_i, E_i, T_1\}$.

According to Message 2, the following statement is obtained.

$S_2 : SM_i \lhd \{ID_j, A_j, B_j, E_j, \sigma_j : \{ID_j, A_j, B_j, T_2, E_j\}_{\{a_j, s_j\}}, T_2\}$.

From S_2, $\mathbb{A}_4$ and R1, the following statement is obtained.

$S_3 : SM_i \mid\equiv (SM_j \mid\sim \{ID_j, A_j, B_j, T_2, E_j\})$.

From S_3, $\mathbb{A}_2$ and R2, the following statement is obtained.

$S_4 : SM_i \mid\equiv (SM_j \mid\equiv \{ID_j, A_j, B_j, T_2, E_j\})$.

From S_4 and R6, the following statement is obtained.

$S_5 : SM_i \mid\equiv (SM_j \mid\equiv E_j)$.

From S_5, $\mathbb{A}_5$, R3, the following statement is obtained.

$S_6 : SM_i \mid\equiv E_j$.

From S_2 and $\mathbb{A}_2$, the following statement is obtained.

$S_7 : SM_i \mid\equiv (\#(E_j))$.

From S_7 and R4, the following statement is obtained due to $k_{i,j} = H(\omega_i \| ID_i \| ID_j)$, $\omega_i = s_i \cdot E_j$.

$S_8 : SM_i \mid\equiv (\#(k_{i,j}))$.

From S_8, S_5 and R5, the following statement is obtained.

$S_9 : SM_i \mid\equiv (SM_i \overset{k_{i,j}}{\leftrightarrow} SM_j)$. (Goal 1)

Then, according to Message 3, the following statement is obtained.

$S_{10} : SM_j \lhd \{A_i, B_i, \sigma_i : \{k_{i,j}, ID_i, ID_j, A_i, B_i, T_3, E_i\}_{\{a_i, s_i\}}, T_3\}$.

From S_{10}, $\mathbb{A}_6$, R1 and R6, the following statement is obtained.

$S_{11} : SM_j \mid\equiv (SM_i \mid\sim \{k_{i,j}, ID_i, ID_j, A_i, B_i, T_3, E_i\})$.

From S_{11}, $\mathbb{A}_2$, R4 and R2, the following statement is obtained.

$S_{12} : SM_j \mid\equiv (SM_i \mid\equiv \{k_{i,j}, ID_i, ID_j, A_i, B_i, T_3, E_i\})$.

From S_{12}, $\mathbb{A}_4$, R3 and R6, the following statements are obtained.

$S_{13} : SM_j \mid\equiv (SM_i \mid\equiv k_{i,j})$.

$S_{14} : SM_j \mid\equiv k_{i,j}$.

From S_{13}, $\mathbb{A}_3$, the following statement is obtained.

$S_{15} : SM_j \mid\equiv (\#(k_{i,j}))$.

From S_{15}, $\mathbb{A}_7$, the following statement is obtained.

$S_{16} : SM_i \mid\equiv (SM_i \overset{k_{i,j}}{\leftrightarrow} SM_j)$. (Goal 2)

The proof is completed.

In the proof, both the nonces E_i and E_j depend on the keys generated by PUF and FE, which aim to provide physical security for devices. The proof result shows that the phase of **Physical Secure Authentication and Blinding Factor Update** achieves the goal of mutual authentication. After the mutual authentication is completed, both SM_i and SM_j can obtain the pairwise factor $k_{i,j}$.

6.3 Informal Security Analysis

This subsection analyzes the security of SERDA informally from three aspects: physical security, security of the pairwise blinding factor, and privacy-preservation.

Theorem 2. (Physical security). If the PUF is considered ideal, SERDA can resist physical attacks such as cloning attacks, extracting secrets from SMs or reading modifications.

Proof. In SERDA, physical attacks are considered in three phases, namely **Physical Secure Authentication and Blinding Factor Update**, **Data Report**, and **Report Aggregation with Fault Tolerance**. During the first two phases, an adversary $\mathcal{A}$ can adjust metering data by altering the configuration of SMs, or easily capture one SM and extract information $\{Ap_i = x_i \oplus a_i, Sp_i = y_i \oplus s_i, Kp_i = H(x_i \| y_i) \oplus k_i, Ek_{i,j} = k_{i,j} \oplus H(a_i)\}$ from the me-

ter and create a clone. Since PUF and FE are integrated in SMs and AG to mask their secret keys, the adversary cannot recover the key K_i by computing $R_i = PUF(C_i)$, $K_i = FE.Rec(R_i, hs_i)$ due to the unclonable nature of PUF. Therefore, in the aforementioned phases, the adversary $\mathcal{A}$ cannot recover the secret keys $\{a_i, s_i, k_i, k_{i,j}\}$ from $\{Ap_i, Sp_i, Kp_i, Ek_{i,j}\}$, which are all masked with the OTPs, which rely on the key K_i generated using the PUF and FE. The same situation occurs when an adversary launches attacks against AG. Moreover, the phase **Physical Secure Authentication and Blinding Factor Update** requires the calculation of $E_i = H(e_i \| K_{i,j}) \cdot P$, $E_j = H(e_j \| K_{j,i}) \cdot P$ where $K_{i,j}$, $K_{i,j}$ are also derived using the PUF and FE integrated in SM_i and SM_j respectively, which actually provides a dual physical security for the phase.

Theorem 3. (Security of the pairwise blinding factor). If the adversary cannot solve CDHP in PPT, then the pairwise blinding factor is secure.

Proof. Assuming an adversary $\mathcal{A}$ attempts to obtain the pairwise blinding factor $k_{i,j}$ of SM_i and SM_j. Here, $k_{i,j} = H(\omega_i \| ID_i \| ID_j)$, $\omega_i = s_i \cdot E_j = s_i \cdot H(e_j \| K_{j,i}) \cdot P$, s_i is the partial private key of SM_i, which is secretly protected using PUF and FE, $s_i \cdot P = H(A_i \| B_i \| ID_i) \cdot PK_C + B_i$ and $E_j = H(e_j \| K_{j,i}) \cdot P$ are public. Therefore, given the tuple $(P, s_i P, H(e_j \| K_{j,i}) \cdot P)$, compute $s_i \cdot H(e_j \| K_{j,i}) \cdot P$ is infeasible in PPT. In other words, the pairwise blinding factor $k_{i,j}$ is secure if the adversary $\mathcal{A}$ cannot solve CDHP in PPT.

Theorem 4. (Privacy-preservation). If the pairwise blinding factors are secure, SERDA can ensure the data privacy of each SM even if CC colludes with AG.

Proof. If CC colludes with AG, it would have the ability to access the encrypted report $c_i = m_i + k_i \pmod q$ of SM_i. However, in order for CC to retrieve the value of the raw data m_i, it must possess knowledge of the blinding factor k_i. Here, k_i is subject to updates using the secure pairwise factor $k_{i,j}$, with each $k_{i,j}$ exclusively known by SM_i and the corresponding SM_j. As a result, even if CC is aware of the original value of k_i, it remains unable to acquire knowledge of the updated blinding factor k_i.

7 Performance Evaluation

In this section, SERDA is compared with three types of data aggregation schemes [13, 18, 27] to demonstrate its efficiency.

7.1 Experimental Setting

To construct a fitting testing environment, a virtual machine configured with 2 virtual CPU cores and 4.00 GB of RAM on a laptop equipped with an AMD R7 5800H CPU @ 3.2 GHz and 16.00 GB of RAM is set up. Subsequently, the Ubuntu 18.04 LTS operating system, along with the Charm-Crypto-0.50 framework [31], the Python-based PUF simulator *pypuf*, and the Python-based FE simulator *python-fuzzy-extractor*, are deployed on the virtual machine to establish the test environment. In the test environment, the bit length of identity and timestamp are both set to 64 bits. Then, the symmetric pairing-friendly curve SS512 ($|G_1| = 512bits$, $|G_t| = 1024bits$, $|Z_q^*| = 160bits$) with a 512-bit base field and embedding degree 2 is selected. Besides, ciphertexts in the BGN cryptosystem are set to 1024 bits in size and the Paillier cryptosystem to 2048 bits in size. Then, we employ the 8-XOR 128-bit XOR Arbiter PUF based on *pypuf*, and simulate the FE with a 160-bit input value using *python-fuzzy-extractor*.

7.2 Computation Cost

The computation cost is measured from three aspects: computation cost on SM, AG, and CC, respectively. We take T_{pa} to denote point addition operation on G_1, T_{sm} to denote scalar multiplication operation on G_1, T_{bp} to

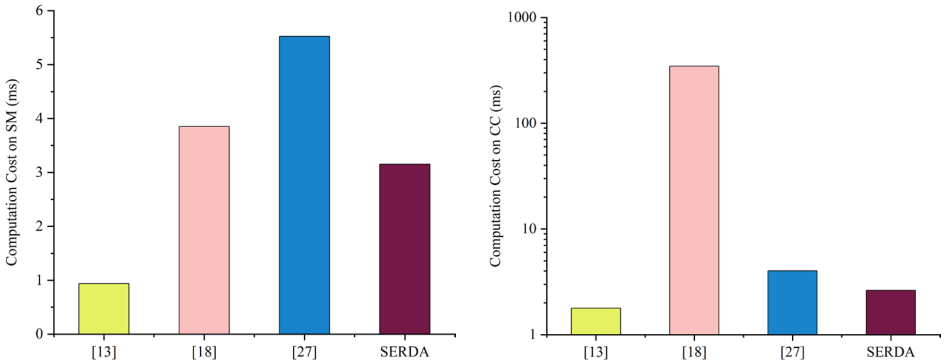
denote the bilinear pairing $\hat{e}: G_1 \times G_1 \to G_T$, T_{h1} to denote hash function that hash an arbitrary bit string to a point on the elliptic curve, T_{h2} to denote hash function that hash an arbitrary bit string to a big integer, T_{puf} to denote PUF operation, and T_{rec} to denote the key recovery algorithm $FE.Rec()$. Besides, we let T_{ge} denote exponentiation operation on $Z_{\hat{n}^2}^*$ in Paillier cryptosystem. Regarding other operations on integer groups, their execution times are so small (less than 0.001ms in simulation) as to be negligible. Therefore, we only take the above operations into consideration in our analysis. This also avoids the impact of multidimensional or multi-subset data structures on computation cost since the techniques used in these structures, such as super-increasing sequence or Chinese Remainder Theorem, rely on modular multiplication and addition operations on integer groups. Then, we set n (the number of SMs) to 100, 200, 300, 400, and 500 for comparison.

In SERDA, each SM_i requires $T_{sm} + 4T_{h2} + T_{puf} + T_{rec}$ to generate the report. Receiving reports from SMs, AG requires $3T_{sm} + 2nT_{pa} + 2nT_{h2}$ to batch verify n pieces of the report data, and $T_{sm} + 2T_{h2} + T_{puf} + T_{rec}$ to generate the aggregated report. Therefore, the computation cost on AG is $4T_{sm} + 2nT_{pa} + 2(n+1)T_{h2} + T_{puf} + T_{rec}$. Then, CC requires $3T_{sm} + 2T_{pa} + 2T_{h2}$ to verify and decrypt the aggregated report.

In scheme [13], each SM_i requires $2T_{ge} + T_{sm} + T_{h2}$ to generate the report. To verify reports from SMs and generate the aggregated report, AG requires $(2n-1)T_{pa} + (n+2)T_{sm} + (n+2)T_{h2} + T_{ge}$. Then, CC requires $T_{pa} + 2T_{sm} + T_{h2} + T_{ge}$ to verify and decrypt the aggregated report.

In scheme [18], each SM_i requires $T_{pa} + 2T_{sm} + 2T_{ge} + T_{h1} + 2T_{h2}$ to generate the report. To verify reports from SMs and generate the aggregated report, AG requires $2(n-1)T_{pa} + 2nT_{sm} + T_{h2} + T_{ge}$. Then, CC requires $nT_{sm} + (n+1)T_{bp} + nT_{h1} + 3T_{h2} + 2T_{ge}$ to verify and decrypt the aggregated report.

In scheme [27], we assume that all SMs generate report with privacy preservation for sake of comparison. Therefore, each SM_i requires $2T_{pa} + 4T_{sm} + T_{h1}$ to generate the report. To verify report from SMs and generate the aggregated report, AG requires $(2n-1)T_{pa} + (n+1)T_{bp} + (n+1)T_{h1} + 2T_{sm}$. Then, CC requires $T_{sm} + 2T_{bp} + T_{h1}$ to verify and decrypt the aggregated report.



(a) SM (b) CC

Fig. 3. Computation cost on (a) SM, (b) CC

The comparison of computation cost among the existing schemes [13, 18, 27] and SERDA is shown in Fig. 3 and Fig. 4. SERDA achieves the minimum total computation cost. Although the computation cost on SM and CC of [13] is lower than SERDA, it comes at the cost of its AG computation cost being much higher than SERDA. Due to the need to verify the reports of each SM on the CC side, the computation cost on CC of [18] is relatively high compared to others. Note that the computation cost of [18] displayed in (c) is for n=100.

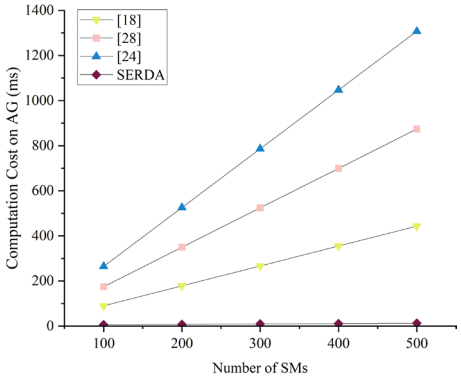


Fig. 4. Computation cost on AG

7.3 Computation Cost

In this subsection, we analyze the communication cost in two aspects, namely, from SM to AG and from AG to CC. For sake of comparison, the fog node number is set to 1 in [13] and [27].

In SERDA, each SM_i sends $\{c_i, A_i, B_i, D_i, ID_i, \delta_i, T_i\}$ to AG. The corresponding communication cost is $(3|G_1|+2|Z_q^*|+|ID_i|+|T_i|)\cdot n = 1984n\ bits$. Then, AG sends $\{c_{sum}, ID_{ag}, D_{ag}, \delta_{ag}, T_{ag}\}$ to CC, with communication cost at $|G_1|+2|Z_q^*|+|ID_{ag}|+|T_{ag}| = 960\ bits$.

In scheme [13], each SM_i sends $\{ID_{S_{ij}}, \delta_{ij}, T_{ij}, c_{ij}\}$ to AG. The corresponding communication cost is $(|ID_{S_{ij}}|+|\delta_{ij}|+|T_{ij}|+|c_{ij}|)\cdot n = 2336n\ bits$. Then, AG sends $\{ID_{ES_i}, \delta_{ES_i}, T_{ES_i}, \widehat{c_i}\}$ to CC, with communication cost at $|ID_{ES_i}|+|\delta_{ES_i}|+|T_{ES_i}|+|\widehat{c_i}| = 2336\ bits$.

In scheme [18], each SM_i sends $\{CT_i, \sigma_i, T\}$ to AG. The corresponding communication cost is $(|CT_i|+|\sigma_i|+|T|)\cdot n = 2624n\ bits$. Then, AG sends $\{CT, \sigma, \xi, T\}$ to CC, with communication cost at $|CT|+|\sigma|+|\xi|+|T| = 3136\ bits$.

In scheme [27], SM_i sends $\{ID_i, TS_i, C_i^{1|2}, \sigma_i^{1|2}\}$ to AG. The corresponding communication cost is $(|ID_i|+|TS_i|+|C_i^{1|2}|+|\sigma_i^{1|2}|)\cdot n = 1152n\ bits$. Then, AG sends $\{ID_f, TS_f, C, \sigma_f\}$ to CC, with communication cost at $|ID_f|+|TS_f|+|C|+|\sigma_f| = 1152\ bits$.

Table 3. Comparison of communication cost

Scheme	[13]	[18]	[27]	SERDA
SM to AG	2336n	2624n	1152n	1984n
AG to CC	2336	3136	1152	960

Table 3 shows the comparison of communication cost among the existing schemes [13, 18, 27] and SERDA. The communication cost of SERDA during the AG to CC phase is minimal, and it is superior to [13] and [18] during the SM to AG phase.

8 Conclusion

In this article, we have proposed the SERDA scheme for smart grid, which can offer enhanced security for key storage and updates while maintaining data privacy and fault tolerance for SM failures. In SERDA, OTPs generated by PUF and FE are utilized to provide enhanced physical security for key storage. Then, an authenticated key agreement protocol based on PUF is designed to achieve mutual authentication and enable SMs involved in the negotiation to verify their physical security implicitly. The security analysis shows that SERDA has achieved the goals of physical security, mutual authentication, and privacy-preservation. The performance evaluation shows that SERDA is efficient compared with related work. In future work, we will delve into data aggregation schemes for scenarios targeting multiple data users.

9 Acknowledgments

This study was supported in part by the Natural Science Foundation of China under grant No. 62072133, in part by the Innovation Project of Guangxi Graduate Education under Grant YCSW2022279, and in part by Wenzhou Science and Technology Plan, No. 2023ZW0013. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M.Z. Gunduz, R. Das, Cyber-security on smart grid: Threats and potential solutions, Computer Networks 169(2020) 107094.
[2] S. Li, K. Xue, Q. Yang, P. Hong, PPMA: Privacy-Preserving Multisubset Data Aggregation in Smart Grid, IEEE Transactions on Industrial Informatics 14(2)(2018) 462-471.
[3] J. Chen, Z. Wang, G. Srivastava, T. A. Alghamdi, F. Khan, S. Kumari, H. Xiong, Industrial blockchain threshold signatures in federated learning for unified space-air-ground-sea model training, Journal of Industrial Information Integration 39(2024) 100593.
[4] P. Kumar, Y. Lin, G. Bai, A. Paverd, J.S. Dong, M. Andrew, Smart Grid Metering Networks: A Survey on Security, Privacy and Open Research Issues, IEEE Communications Surveys & Tutorials 21(3)(2019) 2886-2927.
[5] F. Li, B. Luo, P. Liu, Secure Information Aggregation for Smart Grids Using Homomorphic Encryption, in: Proc. 2010 first IEEE international conference on smart grid communications, 2010.
[6] F.D. Garcia, B. Jacobs, Privacy-Friendly Energy-Metering via Homomorphic Encryption, in Proc. Security and Trust Management: 6th International Workshop, 2010.
[7] R. Lu, X. Liang, X. Li, X. Lin, X. Shen, EPPA: An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communications, IEEE Transactions on Parallel and Distributed Systems 23(9)(2012) 1621-1631.
[8] C.-I. Fan, S.-Y. Huang, Y.-L. Lai, Privacy-Enhanced Data Aggregation Scheme Against Internal Attackers in Smart Grid, IEEE Transactions on Industrial informatics 10(1)(2014) 666-675.
[9] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H.B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical Secure Aggregation for Privacy-Preserving Machine Learning, in: Proc. 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017.
[10] X. Wang, Y. Liu, K.R. Choo, Fault-Tolerant Multisubset Aggregation Scheme for Smart Grid, IEEE Transactions on Industrial Informatics 17(6)(2021) 4065–4072.
[11] Z. Wang, An Identity-Based Data Aggregation Protocol for the Smart Grid, IEEE Transactions on Industrial Informatics 13(5)(2017) 2428–2435.
[12] B. Lang, J. Wang, Z. Cao, Multidimensional data tight aggregation and fine-grained access control in smart grid, Journal of Information Security Applications 40(2018) 156–165.
[13] S. Shang, X. Li, K. Gu, L. Li, X. Zhang, V. Pandi, A Robust Privacy-Preserving Data Aggregation Scheme for Edge-Supported IIoT, IEEE Transactions on Industrial Informatics 20(3)(2024) 4305-4316.
[14] Y. Ding, B. Wang, Y. Wang, K. Zhang, H. Wang, Secure Metering Data Aggregation With Batch Verification in Industrial Smart Grid, IEEE Transactions on Industrial Informatics 16(10)(2020) 6607–6616.
[15] J. Zhang, Y. Zhao, J. Wu, B. Chen, LVPDA: A Lightweight and Verifiable Privacy-Preserving Data Aggregation Scheme for Edge-Enabled IoT, IEEE Internet of Things Journal 7(5)(2020) 4016–4027.
[16] Q. Wu, F.C. Zhou, J. Xu, D. Feng, Lightweight and Verifiable Secure Aggregation for Multi-dimensional Data in Edge-enhanced IoT, Computer Networks 237(2023) 110079.

[17] L. Zhang Y. Liu, FSDA: Flexible Subset Data Aggregation for Smart Grid, IEEE Systems Journal 17(1)(2023) 569–578.
[18] X. Zhang, C. Huang, Y. Zhang, S. Cao, Enabling Verifiable Privacy-Preserving Multi-Type Data Aggregation in Smart Grids, IEEE Transactions on Dependable and Secure Computing 19(6)(2022) 4225–4239.
[19] A. Saleem, A. Khan, S.U.R. Malik, H.B. Pervaiz, H. Malik, M. Alam, A. Jindal, FESDA: Fog-Enabled Secure Data Aggregation in Smart Grid IoT Network, IEEE Internet of Things Journal 7(7)(2020) 6132–6142.
[20] J. Qian, Z. Cao, X. Dong, J. Shen, Z. Liu, Y. Ye, Two Secure and Efficient Lightweight Data Aggregation Schemes for Smart Grid, IEEE Transactions on Smart Grid 12(3)(2021) 2625–2637.
[21] H. Shen, Y. Liu, Z. Xia, M. Zhang, An efficient aggregation scheme resisting on malicious data mining attacks for smart grid, Information Science 526(2020) 289–300.
[22] X. Zhang, C. Huang, C. Xu, Y. Zhang, J. Zhang, H. Wang, Key-Leakage Resilient Encrypted Data Aggregation With Lightweight Verification in Fog-Assisted Smart Grids, IEEE Internet of Things Journal 8(10)(2021) 8234–8245.
[23] Y. Liu, W. Guo, C.-I. Fan, L. Chang, C. Cheng, A Practical Privacy-Preserving Data Aggregation (3PDA) Scheme for Smart Grid, IEEE Transactions on Industrial Informatics 15(3)(2019) 1767–1774.
[24] X. Zuo, L. Li, H. Peng, S. Luo, Y. Yang, Privacy-Preserving Multidimensional Data Aggregation Scheme Without Trusted Authority in Smart Grid, IEEE Systems Journal 15(1)(2021) 395–406.
[25] K. Xue, B. Zhu, Q. Yang, D.S.L. Wei, M. Guizani, An Efficient and Robust Data Aggregation Scheme Without a Trusted Authority for Smart Grid, IEEE Internet of Things Journal 7(3)(2020) 1949–1959.
[26] F. Knirsch, G. Eibl, D. Engel, Error-Resilient Masking Approaches for Privacy Preserving Data Aggregation, IEEE Transactions on Smart Grid 9(4)(2018) 3351–3361.
[27] Z. Zeng, Y. Liu, L. Chang, A Robust and Optional Privacy Data Aggregation Scheme for Fog-Enhanced IoT Network, IEEE Systems Journal 17(1)(2023) 1110–1120.
[28] N. Koblitz, Elliptic curve cryptosystems, Mathematics of computation 48(177)(1987) 203-209.
[29] Y. Liang, E. Luo, Y. Liu, Physically Secure and Conditional-Privacy Authenticated Key Agreement for VANETs, IEEE Transactions on Vehicular Technology 72(6)(2023) 7914–7925.
[30] Y. Dodis, L. Reyzin, A. Smith, Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, in: Proc. Advances in Cryptology-EUROCRYPT 2004: International Conference on The Theory and Applications of Cryptographic Techniques, 2004.
[31] J.A. Akinyele, C. Garman, I. Miers, M.W. Pagano, M. Rushanan, M. Green, A.D. Rubin, Charm: a framework for rapidly prototyping cryptosystems, Journal of Cryptographic Engineering 3(2)(2013) 111–128.