mathematics

MDPI

Article

A Privacy Protection Scheme of Certificateless Aggregate Ring Signcryption Based on SM2 Algorithm in Smart Grid

Hongna Song [1], Zhentao Liu [2], Teng Wang [2], Ling Zhao [2], Haonan Guo [2] and Shuanggen Liu [2,*]

[1] School of Business Administration, Henan Polytechnic University, Jiaozuo 454000, China; songhn@hpu.edu.cn
[2] School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China; 2116200012@stu.xupt.edu.cn (Z.L.); wangteng@xupt.edu.cn (T.W.); zhaoling9543@xupt.edu.cn (L.Z.); ghnxiyou@stu.xupt.edu.cn (H.G.)
* Correspondence: liushuanggen201@xupt.edu.cn

Abstract: With the rapid increase in smart grid users and the increasing cost of user data transmission, proposing an encryption method that does not increase the construction cost while increasing the user ceiling has become the focus of many scholars. At the same time, the increase in users will also lead to more security problems, and it is also necessary to solve the privacy protection for users during information transmission. In order to solve the above problems, this paper proposes an aggregated ring encryption scheme based on the SM2 algorithm with special features, referred to as SM2-CLARSC, based on the certificateless ring signcryption mechanism and combining with the aggregate signcryption. SM2-CLARSC is designed to satisfy the basic needs of the smart grid, and it can be resistant to replay attacks, forward security and backward security, etc. It has better security and higher efficiency than existing solutions. Comparing SM2-CLARSC with existing typical solutions through simulation, the result proves that this solution has more comprehensive functions, higher security, and significant computational efficiency improvement.

Keywords: conditional privacy preservation; aggregate signcryption; certificateless ring signcryption; SM2 algorithm

MSC: 94A60

check for updates

Citation: Song, H.; Liu, Z.; Wang, T.; Zhao, L.; Guo, H.; Liu, S. A Privacy Protection Scheme of Certificateless Aggregate Ring Signcryption Based on SM2 Algorithm in Smart Grid. *Mathematics* **2024**, *12*, 1314. https://doi.org/10.3390/math12091314

Academic Editor: Antanas Cenys

Received: 25 March 2024
Revised: 18 April 2024
Accepted: 22 April 2024
Published: 25 April 2024

1. Introduction

Intelligence is the theme of the development of all walks of life in the future. After combining the traditional grid and the Internet, the smart grid (SG) was developed because the traditional grid can no longer meet the development needs of today's society. In the SG, users interact with the power control center (CC) through smart meters. Users send their own electricity consumption information to CC in real-time, and CC can also feed back information such as billing and predictive pricing to users in SG, allowing users to plan electricity consumption based on feedback, thereby reducing electricity costs. The goal of SG is to provide electricity to users in a more reliable and efficient manner, which has attracted the attention of researchers [1].

However, there are still unresolved issues within the SG. One of the significant challenges pertains to concealing the user's identity within the SG, while also ensuring efficient and rapid completion of signcryption and unsigncryption. At present, many scholars have proposed solutions to this problem, but it can still be improved. The user's private data are easily leaked or damaged during the transmission and storage process with the service node, which has a large security risk. For example, attackers can invade service nodes, which is much less difficult than attacking users or CC [2]. After hijacking edge service nodes, they can reasonably obtain information sent by both users and CC. This is a huge problem that was previously ignored.

At present, the mainstream encryption methods except for regular encryption in SG are ring signcryption [3,4], homomorphic encryption [5,6], etc. These methods cannot prevent service nodes from being hijacked and still protect user privacy and security, as well as data confidentiality. Although attackers cannot break through the algorithm, the data can still be obtained successfully. At the same time, the rapid increase in the number of users in SG will also bring problems such as response delay, service quality degradation, and increased computing pressure on control center resources. Ensuring the confidentiality and unforgeability of user privacy data, while simultaneously addressing response delays due to escalating user numbers poses a significant challenge in Smart Grid (SG) applications. This challenge becomes particularly pronounced when facilitating bidirectional information exchange between users and service nodes.

In response to the above problems, we propose to deploy edge computing nodes (ECN) in the SG in combination with edge computing [7]. ECN simply recalculates the data sent by users. In this process, although ECN receives the ciphertext, it cannot perform complete decryption. ECN can not only perform a simple verification of the ciphertext but also aggregate the ciphertext, so as to improve the computing efficiency of CC and reduce communication costs.

At present, the main methods to address user identity privacy protection issues include ring signature, pseudonym system, and group signature. Among them, adding a pseudonym has higher requirements for the storage cost of SG. The fairness of the group signature mainly depends on the group manager, but this is very subjective. If the system hides the identity of the user completely, it is very likely that malicious users will send malicious information through SG and cannot be found.

After comprehensively studying the existing related schemes, we propose a conditional privacy-preserving ring signcryption scheme based on the SM2 algorithm in a smart grid to address the shortcomings of the existing schemes. In order to effectively solve the user identity privacy protection problem and the problem of computational surge in the control center. The scheme not only outperforms existing related schemes in terms of efficiency but also has the functions of tracking malicious users, aggregating signatures and resisting replay attacks.

1.1. Our Contribution

In general, we propose a new solution. In order to more effectively address the above issues, our research content is as follows:

1. Using the framework of edge computing, it is proposed to alleviate the communication delay problem that may be caused by the surge of users in the SG. ECN partially decrypts and aggregates the ciphertext, and then sends the processed ciphertext to the control center. On the one hand, it can simply verify the ciphertext once, and on the other hand, it can reduce communication costs and improve efficiency.
2. We propose a certificateless aggregate ring signcryption scheme (CLARSC) with conditional privacy protection. This scheme enables the tracking of malicious users while safeguarding the privacy of user identities.
3. Introducing the update key algorithm, which periodically updates the key to prevent greater damage due to the loss of user keys.
4. We compared the scheme proposed in this paper with the existing similar schemes. The results show that the scheme in this paper has more comprehensive functions and significantly improved computational efficiency. By introducing the edge computing structure, the communication pressure of CC is relieved, and the communication cost of the smart grid is reduced.

1.2. Organization

The subsequent sections of this paper are organized as follows: Section 2 offers a review of relevant literature associated with our proposed approach. Section 3 outlines the foundational knowledge. The SM2 signature algorithm is reviewed in Section 4. The certificateless aggregate ring signcryption scheme is introduced in Section 5. Section 6 offers an assessment of the scheme's correctness and security. Performance analysis is detailed in Section 7. Conclusively, Section 8 provides a summary of the key points discussed in this paper.

2. Related Work

The smart grid, as a combination of a traditional power grid and the Internet, began to take shape in the early 2000s. In the beginning, researchers mainly focused on the automation and communication aspects of power systems and paid less attention to privacy issues [8]. The main focus at that time was to enable remote monitoring, control and optimization of power systems [9]. With the introduction of smart meter technology [10], the collection and transmission of user electricity data have become more detailed and frequent. This raises concerns about user privacy [9], especially sensitive information about personal electricity usage behaviors and habits. User privacy issues in smart grids have begun to emerge [11]. Power usage data can reveal users' life patterns and behaviors. Users have expressed concerns that their power usage data and identity information may be abused or leaked. After 2010, many countries and regions began to formulate privacy regulations and policies, requiring power companies to adopt privacy protection measures to ensure the security and privacy of user data. However, there are still criminals who use various methods to obtain users' privacy for illegal profits.

After 2010, researchers mainly used differential privacy algorithms to protect user privacy [12]. In [13], Tian et al. proposed using differential privacy to aggregate multi-data to protect users' data privacy. In [14], Zheng et al. proposed averaging differential privacy to improve the privacy protection performance of the algorithm. Although differential privacy has the characteristics of strong privacy protection, wide applicability, and good standardization, its performance is relatively poor, parameter selection is complicated, and most importantly, it is not suitable for all situations. Moreover, the noise introduced by differential privacy may have a certain impact on the accuracy of data analysis, especially when privacy intensity is high.

In response to the problems of differential privacy, relevant researchers have proposed using ring signatures to protect user identity privacy while improving user experience. The concept of a ring signature, introduced by Rivest et al. in 2001, originated as a simplified form of group signature [15]. The main purpose of ring signatures is to solve the problem of hiding the identity of the real signer during the message transmission process. In [16], Han et al. summarise the issues and problems that have been solved and present approaches that may be able to solve the problems that need to be solved. In their work [17], Wang et al. presented a traceable ring signature scheme designed for batch processing within the SG context. In [18], Tang et al. proposed multi-authority traceable ring signatures for distributed settings in smart grids. Liu et al. proposed an efficient multi-layer linkable ring signature scheme with logarithmic size to address the issue of excessively large signatures, as discussed in [19].

Ring signcryption is proposed as one of the main development directions of ring signatures. In the SG, ring signcryption has attracted much attention because it can encrypt messages while performing ring signatures. Liu et al. presented a trackable ring signature encryption scheme in [20], utilizing the SM2 algorithm. However, this scheme is not suitable for aggregation within smart grid applications. Zhang et al. introduced a ring signcryption scheme in [21], specifically designed to safeguard the privacy of smart meters. In [3], a privacy protection solution for smart meters in decentralized smart homes based on the alliance blockchain is proposed. In [22], Wang et al. proposed a lightweight certificateless aggregation ring signcryption scheme. In [23], Zhang et al. proposed a microgrid point-to-point e-bidding users based on ring signcryption.

The SM2 algorithm is a national cryptography standard [24] proposed by China's National Cryptography Administration (NSA) in 2010 to protect the confidentiality and integrity of information. SM2 is used in various fields because of its high efficiency. In [25], Teng et al. proposed a simple smart grid privacy protection traceability ring signature scheme based on SM2. However, this solution requires signcryption of the message again, and this solution cannot aggregate or batch process signatures, which results in very low efficiency.

Existing ring signcryption schemes in smart grids all have network congestion problems, or cannot simultaneously solve problems such as low efficiency, replay attacks, and attacks on the middler. Therefore, we propose an SM2-based ring signcryption scheme for this aspect, which can solve the above problems at the same time.

3. Preliminary

3.1. Hardness Assumption

- Elliptic Curve Computational Diffie–Hellman Problem (ECCDHP): It pertains to the challenge of efficiently computing the point abP, where G is a known base point on a finite field comprising elliptic curves $E(a,b)$, and aP and bP are given values. This computation cannot be efficiently performed in polynomial time.
- Elliptic Curve Discrete Logarithm Problem (ECDLP): It involves determining the value of x in the equation $Q = xP$, where P and Q are any two points on an additive group $(G,+)$ of order q consisting of points on an elliptic curve $E(a,b)$. This computation cannot be efficiently performed in polynomial time.

3.2. Formal Definition

The scheme comprises eight algorithms, each executed by the following four entities: the Key Generation Center (KGC), Trusted Authority (TRA), as well as ID_s, ID_r, and ID_v.

1. Setup$(1^k) \rightarrow (params, msk, mpk, mtk)$: TRA and KGC cooperate in the performance of this algorithm. The algorithm takes the security parameter k as input and returns the following outputs: the system parameters $params$, the master tracking key mtk, and the master public key mpk.
2. Set-SV$(ID_i) \rightarrow (u_i, U_i)$: The user inputs his identity ID_i and obtains the corresponding secret value u_i and public value U_i.
3. Extract-PSK $(params, U_i) \rightarrow (d_i, V_i)$: KGC executes the algorithm. Entering the system parameters $params$ and U_i, KGC produces the partial private key d_i and the relevant authentication key V_i for the user with the identity ID_i.
4. Generate-PK$(ID_i) \rightarrow (sk_i, PK_i)$: Upon verifying d_i, the user identified as ID_i generates a public-private key pair using d_i and u_i where the private key sk_i and their public key PK_i.
5. Update-Key pairs $(t, ID_i, sk_i^{t-1}, PK_i^{t-1}) \rightarrow (sk_i^t, PK_i^t)$: In the tth cycle, the user with identity ID_i calculates a new key pair using their public-private key pair from the $(t-1)$th cycle. The updated private key is sk_i^t, and the corresponding public key is PK_i^t.
6. Ring Signcryption $(params, sk_s^t, PK_r^t, P_v^t, L, M) \rightarrow (\sigma)$: The user with identity ID_s executes the signcryption algorithm. They use $params, sk_s^t, PK_v^t, L,$ to signcrypt the message M. The output is the signcryption σ.
7. Single Verification $(params, \sigma, L, sk_r^t) \rightarrow (\{0,1\})$: The verifier completes the verification algorithm by inputting $params, \sigma, L$. Additionally, the verifier possesses the private key sk_r^t. Output whether the ciphertext is valid or not
8. Batch Verification $(params, \sigma_{j=1,2,\ldots,m}, sk_r^t) \rightarrow (\{0,1\})$: Enter m signcryptions and sk_r^t of the authenticator ID_r, and prove the correctness of these signcryptions.
9. Aggregated Signcryption $(\sigma_{j=1,2,\ldots,m}, sk_r^t) \rightarrow (\hat{\sigma})$: The ECN ID_r combines m ciphertexts and transmits the aggregated ciphertexts to the control center ID_v.
10. Unsigncryption $(params, \hat{\sigma}, L, sk_v^t) \rightarrow (M_{j=1,2,\ldots,m})$: If the verification result is 1, the verifier uses L and sk_r to unsigncryption $\hat{\sigma}$ and obtain the messages $M_{j=1,2,\ldots,m}$.
11. Tracking $(params, \sigma, \hat{\sigma}) \rightarrow (ID_s)$: When there is a need to track the identity of a malicious signer ID_s. TRA can use the signcryption σ or $\hat{\sigma}$and the ring list L to ascertain the real signer ID_s.

3.3. System Model

As shown in Figure 1, the scheme consists of five main entities: the KGC, the TRA, the Edge Computing Node (ECN), the Control Centre (CC) and the user.

1. KGC: It is responsible for generating partial keys for users, ECNs, and CC.
2. TRA: It is tasked with monitoring the entire power network. In the event of detecting a malicious user, the chase algorithm can be employed to trace the real identity of the signer.
3. ECN: ECN acts as an aggregator in the scheme. It is an edge computing server deployed in the SG which is responsible for processing the ring-encrypted power request information sent by users in a timely manner. The ciphertexts after returning the ring signing encryption are processed and then aggregated to reduce the computation of CC.
4. CC: It is tasked with receiving and verifying the aggregated ciphertext upon receipt, processing the ciphertext to obtain the plaintext, and controlling the power allocation in the SG in real-time in response to the received information.
5. User: The signer in the scheme. Each user $user_i$ sends power usage data to the control center via ECN.

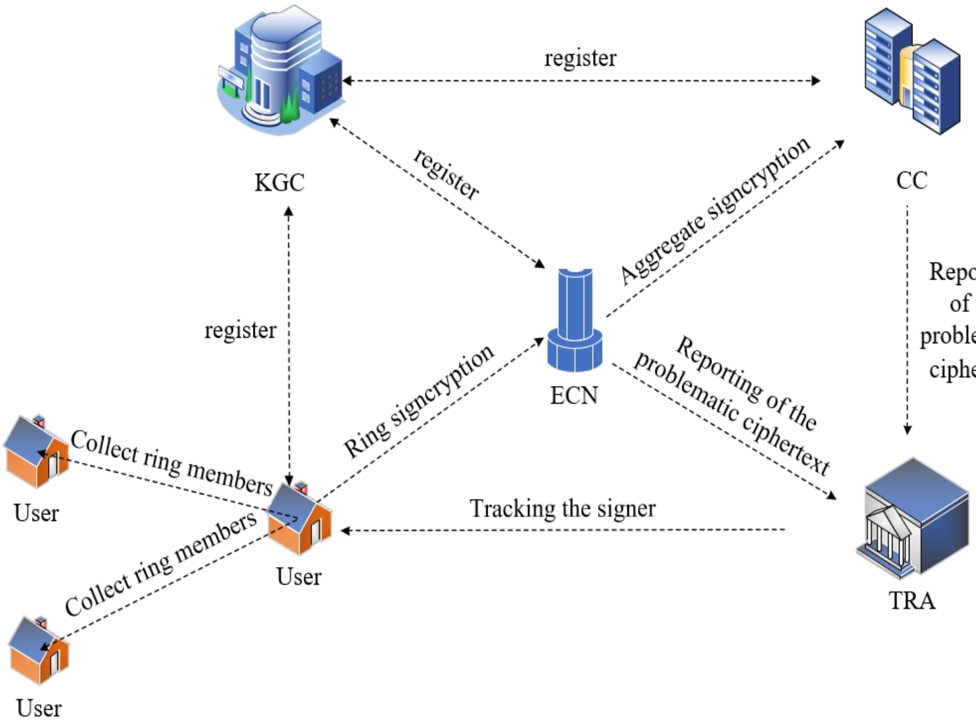

Figure 1. The data transmission architecture of SG.

3.4. Threat Model

The scheme in this paper deals with two types of attacks. The first type of attacker denoted as A_I is one of the ring members. A_I can tamper with any user's public key when generating signature encryption but does not know the system master private key. The second type of attacker is noted as A_{II} is a malicious KGC. A_{II} cannot transform any user's public key but knows the system's master private key. We set up seven Oracle machines for A_I and A_{II} to query as below:

1. Query-H_i: Upon inputting the query value, it can produce the corresponding hash value as output.
2. Query-PSK: Upon entering the ID_i, it can output the corresponding psk_i.
3. Query-SK: If the public key PK_i of the input ID_i is not replaced, the algorithm provides the corresponding private key sk_i.

4. Query-PK: After entering the ID_i, this algorithm outputs the corresponding public key PK_i.
5. Replace-PK: The challenger C inputs the tuple $(ID_i, U^{'})$, and substitutes U_i with $U_i^{'}$.
6. Query-$ARSC$: After entering the tuple (ID_s, ID_r, ID_v, M_j) for $j = 1, \ldots, m$, the challenger C obtains the corresponding ciphertext $\hat{\sigma} = \{\{c_j\}, \{\hat{s_i}\}, \{X_j\}, L, \{I_j\}, \hat{W}, \hat{TS}\}$.
7. Query-USC: By inputting the tuple $(\hat{\sigma}, ID_v)$, the challenger C obtains the decrypted ciphertext $M_{j=1,\ldots,m}$.

Definition 1. *Assuming that the winning advantage of the adversary is negligible in polynomial time in Game 1 and Game 2, the security of the scheme proposed in this paper is for IND-CLRSC-CCA2.*

Proof. Game 1: Opponent A_I and Challenger C participate in the following several phases:

Setting: Challenger C executes the setting algorithm to obtain *params* and then provides them to A_I.

Query: A_I can be queried to the oracle machines and must fulfil the below requirements:

1. A_I cannot perform Query-SK as ID_r, ID_v.
2. A_I cannot perform Query-PSK as ID_r, ID_v, if its public key is replaced.
3. A_I cannot couple the tuple $(\hat{\sigma}, ID_s, ID_r, ID_v)$ to perform the query-USC.

Challenge: A_I outputs two equal length but unique messages M_{j0} and M_{j1}, signer ID_s, ECN ID_r and verifier ID_v, and then forwards them. Challenger C randomly selects $b \in \{0,1\}$ and the tuple $(M_{jb}, ID_s, ID_r, ID_v)$ performs a signed encryption algorithm. Then, C sent $\hat{\sigma}$ to A_I.

Guess: After the adaptive execution of the query phase, A_I guesses $b^{'}$. If $b^{'} = b$, A_I wins this game.

The advantages of A_I are as defined below:

$$Adv_{A_I}^{IND-CLRSC-CCA2} = Pr[A_I\ wins].$$

Game 2: Opponent A_{II} and Challenger C participate in the following several phases:

Setting: C executes the setting algorithm to obtain *params* and then provides them to A_{II}.

Query: A_{II} can be queried to the oracle machines and must fulfil the below requirements:

1. A_{II} cannot perform the Query-SK as ID_r, ID_v.
2. A_{II} cannot perform Query-USC for the tuple $(\hat{\sigma}, ID_s, ID_r, ID_v)$.

Challenge: A_{II} outputs two equal-length but unique messages M_{j0} and M_{j1} with, sender ID_s, ECN ID_r and verifier ID_v, and forwards them. C randomly selects $b \in \{0,1\}$ and uses the tuple $(M_{jb}, ID_s, ID_r, ID_v)$ to execute the signcryption algorithm. Subsequently, C sends $\hat{\sigma}$ back to A_{II}.

Guess: After allowing the query to be executed adaptively in the query stage, A_{II} guesses $b^{'}$. If $b^{'} = b$, A_{II} wins Game 2.

The advantage of A_{II} is defined as follows:

$$Adv_{A_{II}}^{IND-CLRSC-CCA2} = Pr[A_{II}\ wins].$$

□

Definition 2. *Assuming that the winning advantage of the adversary is negligible in polynomial time in Game 3 and Game 4, the security of the scheme proposed in this paper is for EUF-CLRSC-CMA2.*

Proof. Game 3: Opponent A_I and Challenger C participate in the following several phases:

Setting: Same as Game 1.

Query: A_I can be queried to the Oracle machines and must fulfill the below requirements:

1. During the Query-$ARSC$ process, it was unable for A_I to obtain the tuple $(\hat{\sigma}, M_j)$.
2. A_I cannot perform Query-SK as ID_s.

3. If the public key of ID_s has been swapped, A_I could not query Query-PSK.

Forgery: A_I forwards a new tuple $(\hat{\sigma}, M_j, ID_r, ID_v)$. The challenger C uses the tuple $(\hat{\sigma}, M_j, ID_r, ID_v)$ to run the unsigncryption algorithm. If the output of the algorithm is absent, then A_I wins Game 3.

The advantage of A_I is defined as follows:

$$Adv_{A_I}^{EUF-CLRSC-CMA2} = Pr[A_I \ wins].$$

Game 4: Opponent A_{II} and Challenger C participate in the following several phases:

Setting: Same as Game 2.

Query: A_{II} can be queried to the Oracle machines and must fulfill the below requirements:

1. A_{II} cannot perform Query-$ARSC$ on tuple $(\hat{\sigma}, M_j)$.
2. A_{II} cannot perform Query-SK for ID_s.

Forgery: A_{II} inputs a new tuple $(\hat{\sigma}, M_j, ID_r, ID_v)$. The challenger C uses the tuple $(\hat{\sigma}, M_j, ID_r, ID_v)$ to run the unsigncryption algorithm. If the output of the algorithm is absent, then A_{II} wins Game 4.

The advantage of A_{II} is defined as follows:

$$Adv_{A_{II}}^{EUF-CLRSC-CMA2} = Pr[A_{II} \ wins].$$

□

3.5. Security Performance

For better application in SG, this program also has the following properties.

1. Message Validation: The message validator examines the integrity and accuracy of the received data to ensure its integrity and legitimacy as a valid signcryption.
2. Traceability: In the event of malevolent activities within the smart grid, the Traceability mechanism can identify the origin of malicious messages, thereby attributing them to their respective senders.
3. Un-linkability: With the exception of the Traceability mechanism, no entity possesses the ability to discern whether two distinct ciphertexts originate from the same sender.
4. Confidentiality: In order to ensure that unauthorized entities do not have access to the plaintext, it is stipulated that only designated persons can successfully decrypt and access the plaintext.
5. Anonymity: Except for TRA, the sender cannot be traced through analysis of the transmitted message.
6. Replay attack resistance: If an attacker intercepts the ciphertext in the middle of the process, the receiver will consider it to be under attack for as long as the specified time has elapsed.
7. Anti-malicious gateway: By introducing edge computing and aggregate signcryption in ECN, even if malicious nodes want to obtain information, they cannot obtain it.
8. Conditional anonymity: Although ECN and CC can receive the ciphertext, if it is not a malicious user, they cannot know who the specific signcryptor is.
9. User identity privacy protection: During the message-sending process, the user utilizes the ring signcryption algorithm to conceal their identity. This ensures that neither ECN nor CC can determine the source of the information.
10. Forward security: By periodically updating the key, even in the event of accidental private key loss by the user, the security of previously sent messages remains intact and unaffected.

4. Review SM2 Signature Algorithm

This section briefly introduces the general flow of the SM2 digital signature algorithm.

1. System parameter generation: the algorithm inputs security parameter k, and outputs system public parameter $params = \{p, F_p, a, b, P, G, q, H\}$. Where p is a large number, F_p is a finite field. G is the additive cyclic group formed by the points on $E(F_p) : y^2 =$

$x^3 + ax + b \ mod \ p$, its order is q, and P is the base point. $H : \{0,1\}^* \rightarrow Z_q^*$ is a secure hash function.

2. Key generation: User A generates their own $d_A \in Z_q^*$, and calculates $P_A = d_A \cdot P$ as the public key.
3. Signature: A uses d_A to generate a signature for a message m. First, calculates the message digest $e = H(m)$; second, randomly select $k \in Z_q^*$, and computes $(x_1, y_1) = kP$, $r = (e + x_1) \ mod \ q$, $s = [(1 + d_A)^{-1} \cdot (k - r \cdot d_A)]$. Finally output the signature (r,s).
4. Verification: After receiving the message m' and the signature (r', s'), the verifier first checks whether $r', s' \in Z_q^*$ is true. If true, the verification calculates $e' = H(m')$, $t = (r' + s')$. Then they can use s' and t to compute $(x_1', y_1') = s'P + tP_A$, and calculate $R = (e' + x_1') \ mod \ q$. Then, verify whether the equation $R = r'$ is true, if false, (r,s) is an invalid signature about m, otherwise the signature is valid.

5. SM2-Based Certificateless Aggregate Ring Signcryption Scheme

In this section, we present the detailed design for the SG and provide the corresponding symbols, which are listed in Table 1 for reference. The operation process of certificateless aggregate ring signcryption is shown in Figure 2. The specific operation process is as follows:

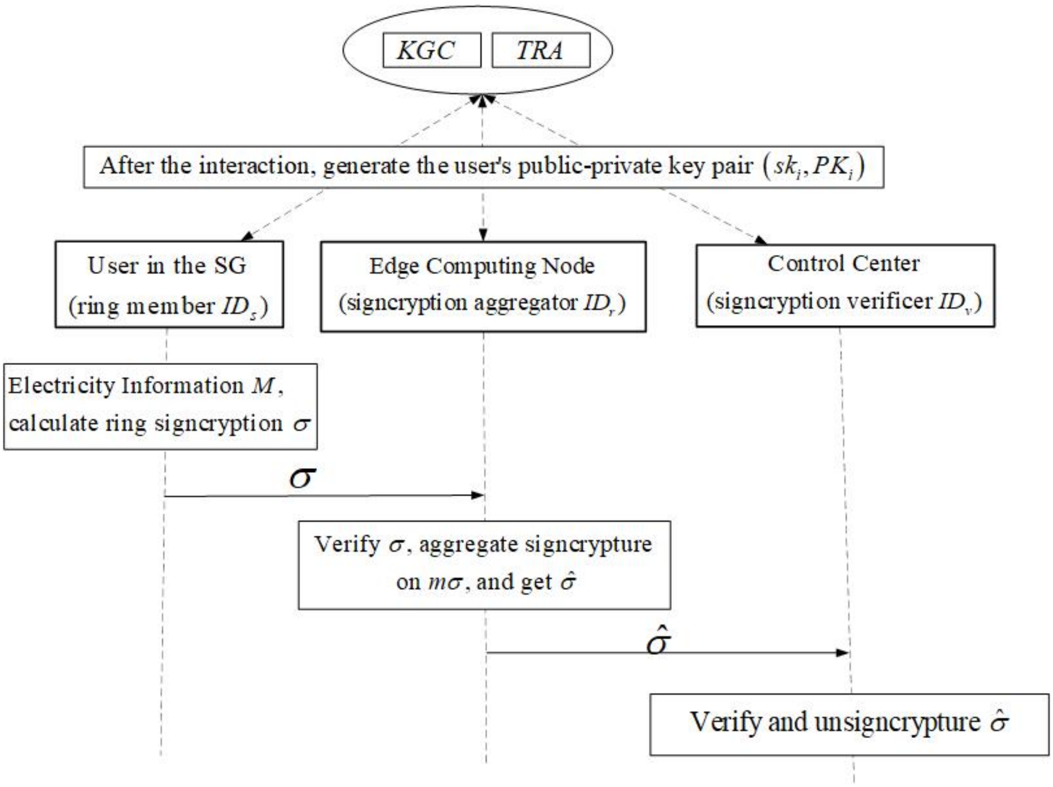
KGC
TRA
After the interaction, generate the user's public-private key pair (sk_i, PK_i)
User in the SG (ring member ID_s)
Edge Computing Node (signcryption aggregator ID_r)
Control Center (signcryption verificer ID_v)
Electricity Information M, calculate ring signcryption σ
σ
Verify σ, aggregate signcrypture on $m\sigma$, and get $\hat{\sigma}$
$\hat{\sigma}$
Verify and unsigncrypture $\hat{\sigma}$

Figure 2. The process of running the program in this paper.

Table 1. Symbols and their Meanings.

Notations	Meanings
k	Security parameter
P	The generator of G
G	Additive group
H_1, H_2, H_3, H_4, H_5	Hash function
T_{pub}	Public key of TRA

Table 1. *Cont.*

Notations	Meanings
ID_i	The identity of $user_i$
d_i	The partial private key of user ID_i
sk_i	The private key of ID_i
PK_i	The public key of ID_i
M	Awaiting Signcrypted Messages
L	Public key collection
I	Tracking mark
TS	Timestamp
σ	Signcrypted ciphertext

Below delineates the implementation process of our proposed program:

1. **Setup**: To execute the following steps, input the security parameter k, KGC, and TRA:
   (a) KGC chooses two large prime numbers p and q such that $p, q > 2^k$ and a finite field F_p. The equation of an elliptic curve $E : y^2 = x^3 + ax + b \ mod \ p$ defined on F_p. Points satisfying this equation form an abelian group G of order q with base point P.
   (b) The KGC randomly selects $x \in Z_q^*$ as the master private key msk and computes $P_{pub} = xP$ as the master public key mpk.
   (c) The KGC sets up security hash functions H_1, H_2, H_3, and H_4 as follows: $H_1 : \{0,1\}^* \times G \to Z_q^*$, $H_2 : \{0,1\}^* \to \{0,1\}^l$, $H_3 : \{0,1\}^* \to Z_q^*$, $H_4 : \{0,1\}^* \times \{0,1\}^l \to \{0,1\}^l$, $H_5 : \{0,1\}^* \times G \to Z_q^*$. The length of the message is l.
   (d) TRA randomly chooses $t \in Z_q^*$ and calculates $T_{pub} = tP$.
   (e) KGC and TRA publishes params: $params = \{p, q, G, P, P_{pub}, T_{pub}, H_1, H_2, H_3, H_4, H_5\}$.
2. $\mathbf{Set - SV}$: The user ID_i randomly selects $u_i \in Z_q^*$ and computes $U_i = u_i P$. Subsequently, U_i is sent to the key generation center (KGC).
3. $\mathbf{Extract - PSK}$: Upon receiving U_i, the KGC randomly selects $v_i \in Z_q^*$ and calculates $V_i = v_i P$. Then, it calculates $e_i = H_1(ID_i, U_i, V_i, P_{pub})$ and $d_i = v_i + e_i x$, where the partial private key is denoted by d_i. KGC exposes V_i and sends $D_i = (d_i, V_i)$ to ID_i.
4. $\mathbf{Generate - PK}$: The user ID_i acquires D_i and tests the validity of d_i using the formula: $d_i P = V_i + H_1(ID_i, U_i, V_i, P_{pub}) P_{pub}$.
   If it is not, the user will recalculate the key. If the equation holds, ID_i will be given a partial privy d_i and the current period's privy will be calculated $sk_i^1 = u_i + d_i \ mod \ q$. Consequently, the corresponding public key is set as $PK_i^1 = sk_i^1 P$.
5. $\mathbf{Update - Key\ Pairs}$: During the t-th cycle, the user ID_i randomly generates a number $u_i^t \in Z_q^*$. The updated private key is calculated as $sk_i^t = sk_i^{t-1} + u_i^t$, and the corresponding public key is computed as $PK_i^t = sk_i^t P$. The updated public key PK_i^t is then delivered.
6. **Ring Signcryption**: ID_s encrypts the message M using the ring public key $L = \{ID_1, ID_2, \ldots, ID_n\}$ and PK_r of the ECN ID_r, PK_v of CC ID_v, and finish the steps below.
   (a) ID_s randomly selects $d \in Z_q^*$, and computes $X = (x_1, y_1) = d \cdot P$, $Y = (x_2, y_2) = d \cdot PK_r^t$, $Z = (x_3, y_3) = d \cdot PK_v^t$.
   (b) ID_s performs the following calculations, where M is the message to be signed, I is the tracking tag and $\oplus$ is the XOR operator:

$$c = H_2(x_3||y_3) \oplus M \tag{1}$$

$$C = H_3(x_2||y_2) \oplus c \tag{2}$$

$$\beta = H_4(x_1||c||y_1) \tag{3}$$

$$I = (sk_s^t \cdot \beta) T_{pub} \tag{4}$$

$$r_i = H_5(L, c, X, I)(i = 1, 2, \dots, n) \tag{5}$$

(c) Randomly select numbers $k, s_i \in Z_q^*(i = 1, \dots, s-1, s+1, \dots, n)$. Calculates: $Q = (\sum_{i=1, i \neq s}^{n} s_i)P + \sum_{i=1, i \neq s}^{n}[(r_i + s_i)PK_i^t]$, $W = kP + Q$.

(d) ID_s computes $s_s = [(1 + sk_s^t)^{-1}(k - r_s sk_s^t)]\ mod\ q$.

(e) Add a timestamp TS to σ. Then, send σ to ID_r: $\sigma = \{C, \{s_i\}, X, L, I, W, TS\}$.

7. **Single Verification**: We denote the received ciphertext by $\sigma = \{C, \{s_i\}, X, L, I, W, TS\}$. Upon receiving the ciphertext σ, the receiver ID_r performs the following calculation to verify its validity.

   (a) The receiver ID_r verifies the validity of TS using the formula $|TS - TS_{cur}| \leq \triangle TS$, where $\triangle TS$ denotes the maximum acceptable time interval and TS_{cur} represents the current timestamp.

   (b) ID_r checks whether $s_i \in Z_q^*$ for $i = 1, 2, \dots, n$. If any of the s_i values are not in Z_q^*, ID_r discards the message.

   (c) The receiver ID_r computes $Y' = (x_2', y_2') = sk_r^t \cdot X$, $c' = H_3(x_2'||y_2') \oplus C$, and $r_i' = H_5(L, c', X, I), (i = 1, 2, \dots, n)$.

   (d) ID_r checks whether $W' = (\sum_{i=1}^{n} s_i)P + \sum_{i=1}^{n}[(r_i' + s_i)PK_i^t]$. If the equation holds true, ID_r is assured that the ciphertext σ is real and proceeds to receive the message. If the equation does not hold, ID_r reports to TRA and discards it.

8. **Batch Verification**: Perform batch verification on messages

$$\sigma_j = \{C_j, \{s_i\}_j, X, L, I_j, W_j, TS_j\}(j = 1, 2, \dots, m).$$

   (a) Check whether $s_{ij} \in Z_q^*$ for $(1 \leq i \leq n, 1 \leq j \leq m)$.

   (b) The receiver ID_r computes the following values:

$$Y' = (x_2', y_2') = sk_r^t \cdot X \tag{6}$$

$$c_j' = H_3(x_2'||y_2') \oplus C_j \tag{7}$$

$$r_{ij}' = H_5(L, c_j', X, I_j) for (1 \leq i \leq n, 1 \leq j \leq m) \tag{8}$$

$$W'' = (\sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij})P + \sum_{i=1}^{n}[\sum_{j=1}^{m}(r_{ij}' + s_{ij})PK_i^t] \tag{9}$$

   ID_r needs to check if $W'' = \sum_{j=1}^{m} W_j$. If they are equal, ID_r can be certain that the ciphertexts $\sigma_1, \sigma_2, \dots, \sigma_m$ are correct and can receive them.

9. **Aggregated Signcryption**: ID_r aggregates m signcryptions, where the encrypted information is: $M_j(1 \leq j \leq m)$.

   (a) The receiver ID_r performs the following computations: $Y' = (x_2', y_2') = sk_r^t \cdot X$, $c_j' = H_3(x_2'||y_2') \oplus C_j$,

   (b) Compute $\hat{s_i} = \sum_{j=1}^{m} s_{ij}$, $\hat{W} = \sum_{j=1}^{m} W_j$,

   (c) Generate a new timestamp $\hat{TS}$

   (d) Perform the aggregated signcryption as follows

$$\hat{\sigma} = \{\{c_j'\}, \{\hat{s_i}\}, X, L, \{I_j\}, \hat{W}, \hat{TS}\}.$$

10. **Aggregated Verification**: ID_r aggregates m signcryptions, where the encrypted information is: $M_j(1 \leq j \leq m)$.

   (a) ID_v checks $|\hat{TS} - TS_{cur}| \leq \triangle TS$.

   (b) ID_v needs to verify $\hat{s_i} \in Z_q^*(1 \leq i \leq n)$.

   (c) The receiver ID_v computes $r_{ij}'' = H_5(L, c_j', X, I_j)$ for $j = 1, \dots, m$, $\hat{W}' = (\sum_{i=1}^{n} \hat{s_i})P + \sum_{i=1}^{n}[(\sum_{j=1}^{m} r_{ij}'' + \hat{s_i})PK_i^t]$.

   (d) ID_v needs to check $\hat{W}' = \hat{W}$.

(e) ID_v then restores the encrypted message through the following calculation: $Z' = (x_3', y_3') = sk_v^t \cdot X$, $M_j' = H_2(x_3'||y_3') \oplus c_j$.

11. **Tracking**: In instances where the message fails the verification process, ID_v has the discretion to escalate the matter to TRA. Additionally, TRA monitors for malicious activity in the SG. When a malicious ciphertext is found TRA can utilize the equation for $k^{-1}I = H_4(x_1||c||y_1) \cdot PK_j^t$ to ascertain the malicious user ID_j from the ring set $L = \{ID_1, ID_2, \ldots, ID_n\}$.

6. Safety Analysis

6.1. Proof of Correctness

In this section, we present a comprehensive analysis of the security of the aforementioned scheme.

For $i = 1, 2, \ldots, n$,

$$r_i' = H_5(L, c', X, I) \tag{10}$$

$$\begin{aligned}
W' &= (\sum_{i=1}^{n} s_i)P + \sum_{i=1}^{n}[(r_i' + s_i)PK_i^t] \\
&= s_sP + (r_s' + s_s)PK_i^t + (\sum_{i=1, i\neq s}^{n} s_i)P + \sum_{i=1, i\neq s}^{n}[(r_i + s_i)PK_i^t] \\
&= s_sP + (r_s' + s_s)PK_i^t + Q \\
&= s_s(P + PK_i^t) + r_s'PK_i^t + Q \\
&= (1 + sk_s^t)^{-1}(k_s - r_ssk_s^t)(P + PK_i^t) + r_s'PK_i^t + Q \\
&= (1 + sk_s^t)^{-1}(k_s - r_ssk_s^t)(1 + sk_s^t)P + r_s'PK_i^t + Q \\
&= (k - r_ssk_s^t)P + r_s'PK_i^t + Q \\
&= kP + Q \\
&= W
\end{aligned} \tag{11}$$

Aggregated verification:

$$\begin{aligned}
\hat{W}' &= (\sum_{i=1}^{n} \hat{s}_i)P + \sum_{i=1}^{n}[(\sum_{j=1}^{m} r_{ij}'' + \hat{s}_i)PK_i^t] \\
&= (\sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij})P + \sum_{i=1}^{n}[(\sum_{j=1}^{m} r_{ij}'' + \sum_{j=1}^{m} s_{ij})PK_i^t] \\
&= \sum_{j=1}^{m}(\sum_{i=1}^{n} s_{ij})P + \sum_{i=1}^{n}[(r_i''' + s_{ij})PK_i^t] \\
&= \sum_{j=1}^{m} W_j \\
&= \hat{W}
\end{aligned} \tag{12}$$

Unsigncryption:

For message M_j and its encrypted ciphertexts is

$$\sigma_j = \{C_j, s_{ij}, X_j, L, I_j, W_j, TS_j\} \tag{13}$$

$$\hat{\sigma} = \{\{c_j'\}, \{\hat{s}_i\}, X, L, \{I_j\}, \hat{W}, \hat{TS}\} \tag{14}$$

$$Y' = (x_2', y_2') = sk_r^t \cdot X \tag{15}$$

$$Z' = (x_3', y_3') = sk_r^t \cdot X \tag{16}$$

$$\begin{aligned} M'_j &= H_2(x'_3||y'_3) \oplus c'_j \\ &= H_2(x'_3||y'_3) \oplus H_3(x'_2||y'_2) \oplus C_j \\ &= H_2(x'_3||y'_3) \oplus H_3(x'_2||y'_2) \oplus H_3(x_2||y_2) \oplus c_j \\ &= H_2(x'_3||y'_3) \oplus H_3(x'_2||y'_2) \oplus H_3(x_2||y_2) \oplus H_2(x_3||y_3) \oplus M_j \\ &= M_j \end{aligned} \tag{17}$$

Tracking:

$$k^{-1}I = H_4(x_1||c||y_1)PK_j^t \tag{18}$$

$$k^{-1}(sk_j^t \cdot \beta)T_{pub} = \beta PK_j^t \tag{19}$$

$$(sk_j \cdot \beta)P = \beta \cdot PK_j^t \tag{20}$$

Based on the above verification, we can conclude that the scheme proposed in this paper is both correct and reasonable. In the following sections of this chapter, we will provide proof to establish the security, and functionality of this scheme.

6.2. Confidentiality

Theorem 1. *If a Type I adversary A_I manages to achieve a non-negligible advantage ε in Game 1, successfully compromising IND-CLRSC-CCCA2, after executing q_{H_i} queries to Query-H_i (for $i = 1,2,3,4,5$), q_{PSK} queries to Query-PSK, q_{SK} queries to Query-SK, q_{PK} queries to Query-PK, q_{RPK} queries to Replace-PK, q_{ARSC} queries to Query-ARSC, and q_{USC} queries to Query-USC, then the Elliptic Curve Computational Diffie–Hellman Problem (ECCDHP) can be resolved with a probability $\varepsilon' \geq \varepsilon(1 - q_{USC}/2^l)/[e(q_{PSK} + q_{SK} + q_{RPK})]$, where l is the length of the signcryption message, and e denotes the base of the natural logarithm.*

Proof. Assume the challenger C is given the tuple $(P, aP, bP) \in G^3$ and is tasked with computing the value of abP. In Game 1, C acts as the simulator while A_I acts as the adversary. We set $Pr(ID_i = ID^*) = \delta$, where ID^* represents the target identity.

Setup: C performs the setup, obtaining *params* and $P_{pub} = aP$. Then, C transmits *params* to A_I.

Query: C simulates the oracles as follows for A_I and maintains the lists: L_1, L_2, L_3, L_4, L_5, L_U, L_{PK}, L_{PSK}, L_{SK} and those lists are empty initially.

Query $- H_1$: When A_I provides the tuples $(ID_i, T_i, R_i, P_{pub})$, C checks the list L_1 for related tuples.

1. If $(ID_i, U_i, R_i, P_{pub}, e_i) \in L_1$, C obtains e_i from L_1 and feedback e_i to A_I.
2. If $(ID_i, U_i, R_i, P_{pub}, e_i) \notin L_1$, C random chooses a number $e_i \in Z_q^*$ and return e_i to the enemy A_I and C stores $(ID_i, U_i, R_i, P_{pub}, e_i)$ into the list L_2.

Query $- H_2$: When A_I receives the tuple (x_3, y_3), component C searches the list L_3 for a tuple that is related to it.

1. If $(x_3, y_3, h_2) \in L_2$, C gains h_2 from L_2 and feedback h_2 to A_I.
2. If $(x_3, y_3, h_2) \notin L_2$, C random chooses a number $h_2 \in Z_q^*$, feedbacks h_2 to A_I and stores (x_3, y_3, h_2) into the list L_2.

Query $- H_3$: When A_I provides the tuple (x_2, y_2), C Examine list L_3 for related tuples.

1. If $(x_2, y_2, h_3) \in L_3$, C obtains h_3 from L_3 and sets h_3 as a reply to A_I.
2. If $(x_2, y_2, h_3) \notin L_3$, C chooses $h_3 \in Z_q^*$ randomly, sends out h_3 to A_I and stores (x_2, y_2, h_3) into the list L_3.

Query $- H_4$: When A_I provides the tuple $(x_1||c||y_1)$, C Examine list L_4 for related tuples.

1. If $(x_1||c||y_1, \beta) \in L_4$, C derive β from L_4, responses β to A_I.
2. If $(x_1||c||y_1, \beta) \notin L_4$, C randomly chooses $\beta \in Z_q^*$, responses β to A_I and stores $(x_1||c||y_1, \beta)$ into the list L_4.

Query $- H_5$: When A_I supplies the tuples (L, c_j, X, I_j), C Examine list L_5 for related tuples.

1. If $(L, c_j, X, I_j, r_i) \in L_5$, C derive r_i from L_5, responses r_i to A_I.
2. If $(L, c_j, X, I_j, r_i) \notin L_5$, C randomly chooses $r_i \in Z_q^*$, responses r_i to A_I and stores (L, c_j, X, I_j, r_i) into the list L_5.

$Query - PSK$: When A_I requests the partial private key for identity ID_i, C checks the list L_{PSK}.

1. If $(ID_i, d_i) \in L_{PSK}$, C sends d_i to A_I.
2. If $(ID_i, d_i) \notin L_{PSK}$, and $ID_i \neq ID^*$, C randomly selects $d_i \in Z_q^*$ and returns it to A_I. C then adds the tuple (ID_i, d_i) to L_{PSK}. If $ID_i = ID^*$, C fails.

$Query - SK$: When A_I requests the private key for identity ID_i, C checks the list L_{SK}.

1. If $(ID_i, sk_i) \in L_{SK}$, C returns sk_i to A_I.
2. If $(ID_i, sk_i) \notin L_{SK}$, and $ID_i \neq ID^*$, C searches for the relative tuples (ID_i, u_i) and (ID_i, d_i) from the lists L_U and L_{PSK} to obtain u_i, d_i. C then computes $sk_i = d_i + u_i$ and returns it to A_I. Additionally, C adds the tuple (ID_i, sk_i) to L_{SK}. If $ID_i = ID^*$, C fails.

$Query - PK$: When A_I requests the public key for identity ID_i, C checks the list L_{PK}.

1. If $(ID_i, PK_i) \in L_{PK}$, C searches for (ID_i, PK_i) in L_{PK} and returns PK_i to A_I.
2. If $(ID_i, PK_i) \notin L_{PK}$, and $ID_i = ID^*$, C randomly selects numbers $v_i, u_i \in Z_q^*$, and computes $e_i = H_1(ID_i, U_i, V_i, P_{pub})$. C then sets $PK^* = PK_i = (u_i + v_i + e_i a)P$ as a response to A_I. Afterward, C stores (ID_i, u_i), $(ID_i, U_i, R_i, P_{pub}, e_i)$ into the lists L_U and L_1, respectively. If $ID_i \neq ID^*$, C randomly selects numbers $v_i, d_i \in Z_q^*$, computes $PK_i = (u_i + d_i)P$, and returns PK_i to A_I. Subsequently, C adds the tuple (ID_i, PK_i) into the list L_{PK}.

$Replace - PK$: When A_I relays the tuple $(ID_i, PK_i^{t'})$, C updates the tuple $(ID_i, PK_i^{t'})$ with (ID_i, PK_i^t) in the L_{PK}.

$Query - ARSC$: Assuming it is the t-th cycle, and A_I relays the tuple $(ID_s, ID_r, ID_v, M_{j=1,\ldots,m})$, for any message M_j in this tuple, C performs the following operations.

1. If $ID_s = ID^*$ and $ID_r \neq ID^*$:
   (a) C randomly selects a point $I_j \in G$, queries Query-PK for ID_r and ID_v, respectively, and obtains PK_r^t and PK_v^t.
   (b) C randomly selects a value $d_j \in Z_q^*$, and computes $X = (x_1, y_1) = d \cdot P$, $Y = (x_2, y_2) = d \cdot PK_r^t$, $Z = (x_3, y_3) = d \cdot PK_v^t$.
   (c) C computes $c_j = H_2(x_3, y_3) \oplus M_j$, $C_j = H_2(x_2 || y_2) \oplus c_j$, and $r_{ij} = H_4(L, c_j, X, I_j)$ for $i = 1, 2, \ldots, n$.
   (d) C randomly selects figures $s_{ij} \in Z_q^*(i = 1, \ldots, s-1, s+1, \ldots, n)$, maths $W' = (\sum_{i=1}^{n} s_i)P + \sum_{i=1}^{n}[(r_{ij} + s_{ij})PK_i^t]$, $\hat{s_i} = \sum_{i=1}^{n} s_i$.
   (e) C applies the Aggregated Signcryption algorithm and obtains a new timestamp $\hat{TS}$.
   (f) C sends the ciphertext
   $\hat{\sigma} = \{\{c_j\}, \{\hat{s_i}\}, X, L, \{I_j\}, \hat{W}, \hat{TS}\}$
   to A_I, and stores the tuples $(x_2 || y_2, h_2)$ and $(x_3 || y_3, h_3)$ into the list L_3 and L_2, and stores the tuples (L, c_j, X, I_j, r_{ij}) into the list L_5.
2. If $ID_r = ID^*$ and $ID_s \neq ID^*$:
   (a) C applies the Ring Signcryption algorithm.
   (b) For all message ciphers $\sigma_j (j = 1, \ldots, n)$, C computes $Y = (x_2, y_2) = d_j \cdot PK_r^t$, $c_j = H_2(x_2 || y_2) \oplus C_j$, $\hat{s_i} = \sum_{j=1}^{m} s_{ij}$, and $\hat{W} = \sum_{j=1}^{m} W$.
   (c) C sends the ciphertext
   $\hat{\sigma} = \{\{c_j\}, \{\hat{s_i}\}, X, L, \{I_j\}, \hat{W}, \hat{TS}\}$ to A_I.
3. If $ID_s \neq ID^*$ and $ID_r \neq ID^*$: C apply both the Ring Signcryption Algorithm and the Aggregated Signcryption Algorithm.

$Query - USC$: A_I relays $\hat{\sigma} = \{\{c_j\}, \{\hat{s_i}\}, X, L, \{I_j\}, \hat{W}, \hat{TS}\}$ and an identity ID_v:

1. If $ID_v = ID^*$, C searches the relative tuples $(x_3||y_3, h_2)$ and $(x_2||y_2, h_3)$ from the list L_2,L_3. Finds the tuples (L, c_j, X, I_j, r_{ij}) from the list L_4. If these tuples are absent, C rejects σ. Otherwise, C runs the Verification algorithm and calculates $M_j = h_{3j} \oplus c_j$. C then returns M_j to A_I for $j = 1, \ldots, m$.
2. If $ID_v \neq ID^*$, C runs the Unsigncryption method.

Challenge: A_I selects two distinct messages, denoted as M_0 and M_1, which are of equal length. Additionally, A_I chooses a sender as ID_s, the ECN as ID_r, and an acceptor as ID_v. These messages, along with ID_s, ID_r, and ID_v, are forwarded to C along with the identities of the ring members as $L = \{ID_1, ID_2, \ldots, ID_n\}$.

1. If $ID_v = ID^*$, C randomly selects a bit $b \in \{0, 1\}$ and performs the following process:
   (a) C sets $X = b \cdot P$, $Y = b \cdot PK_r^t$, $Z = b \cdot PK_v^t$.
   (b) C computes $c_j = H_2(x_3||y_3) \oplus M_j$, $C_j = H_3(x_2||y_2) \oplus c_j$, $\beta = H_3(x_1||c_j||y_1)$, $I_j = (sk_s^t \cdot \beta)T_{pub}$ and $r_i = H_5(L, c_j, X, I_j)$ for $j = 1, \ldots, m$.
   (c) C randomly selects figures $\hat{s}_i \in Z_q^*$ for $i = 1, 2, \ldots, n$, and computes $\hat{W} = (\sum_{i=1}^{n} \hat{s}_i)P + \sum_{i=1}^{n}[(\sum_{j=1}^{m} r_{ij} + \hat{s}_i)PK_i^t]$.
   (d) C increases the timestamp TS to ciphertext $\hat{\sigma}$ and returns $\hat{\sigma}$ to A_I.

$$\hat{\sigma} = \{\{c_j\}, \{\hat{s}_i\}, X, L, \{I_j\}, \hat{W}, \hat{TS}\}$$

2. If $ID_v \neq ID^*$, C fails.

Guess: A_I executes adaptive querying, and guesses b'. If A_I relays the tuples (x_3, y_3, h_2) to Query-H_2, it would know that $\hat{\sigma}$ is a flawed ciphertext. Then, C can solve the ECCDHP that $abP = e_v^{-1}[Z - (u_v + v_v)X]$.

We define the following two cases:

π_1: C passing the query stage.

π_2: C passing the challenge stage.

We can deduce that:

$$\begin{aligned} Pr[\pi_1] &= (1-\delta)^{q_{PSK}+q_{SK}+q_{RPK}}(1 - \frac{q_{USC}}{2^l}), \\ Pr[\pi_2|\pi_1] &= \delta, \\ Pr[C\ success] &= Pr[\pi_1 \wedge \pi_2] \\ &= Pr[\pi_1]Pr[\pi_2|\pi_1] \\ &= (1-\delta)^{q_{PSK}+q_{SK}+q_{RPK}} \cdot \delta(1 - \frac{q_{USC}}{2^l}) \\ &\geq \frac{1 - \dfrac{q_{USC}}{2^l}}{e(q_{PSK} + q_{SK} + q_{RPK})} \end{aligned} \tag{21}$$

where $\delta = \frac{1}{q_{PSK}+q_{SK}+q_{RPK}+1}$.

Thus, C can be used with probability $\varepsilon' \geq \varepsilon(1 - q_{USC}/2^l)/[e(q_{PSK} + q_{SK} + q_{RPK})]$ to solve the ECCDHP, if A_I's advantage of success is ε. □

Theorem 2. *If a Type II opponent A_{II} can achieve successfully attack IND-CLRSC-CCA2 for a non-negligible advantage ε in Game 2, algorithm C with a probability $\varepsilon' \geq \varepsilon(1 - q_{USC}/2^l)/(eq_{SK})$ can be solved the ECCDHP.*

Proof. Let us assume that the simulator C obtains the tuple $(P, aP, bP) \in G^3$ and its task is to compute the value of abP. The simulator is C and the adversary isA_{II} in Game II. Set $Pr(ID_i = ID^*) = \delta$.

Setup: C executes the Setup in Section 3 and generates the system parameters $params = \{p, q, G, P, P_{pub}, T_{pub}, H_1, H_2, H_3, H_4, H_5\}$. C then computes $P_{pub} = xP$ and sends the $params$ to the adversary A_{II}.

Query: C and upholds the initially empty lists L_1, L_2, L_3, L_4, L_5, L_U, L_{PK}, L_{PSK} and L_{SK}, which are initially empty.

$Query - PSK$: When A_{II} relays an identity ID_i:

1. If $(ID_i, d_i) \in L_{PSK}$, C sends d_i to A_{II}.
2. If $(ID_i, d_i) \notin L_{PSK}$, and $ID_i \neq ID^*$, C randomly selects a number $v_i \in Z_q^*$, searches for e_i from the tuples $(ID_i, U_i, V_i, P_{pub}, e_i)$ in the list L_1, and computes $d_i = v_i + e_i x$. C then sends d_i to A_{II}. If $ID_i = ID^*$, C fails.

$Query - PK$: At the i-th query, C sets a challenger identity $ID_i = ID^*$. When A_{II} submits an identity ID_i:

1. If $(ID_i, PK_i) \in L_{PK}$, C searches (ID_i, PK_i) in L_{PK} and returns PK_i to A_{II}.
2. If $(ID_i, PK_i) \notin L_{PK}$, and $ID_i = ID^*$, C randomly selects a number $d_i \in Z_q^*$, and sets $PK^* = PK_i = (d_i + a)P$ as a response to A_{II}. If $ID_i \neq ID^*$, C randomly selects numbers $d_i, v_i \in Z_q^*$, sets $PK^* = PK_i = (u_i + d_i)P$, and responds with PK_i to A_{II}. Then, C buffers the tuples (ID_i, PK_i) and (ID_i, u_i) into the list L_{PK} and L_U, respectively.

Other query types remain the same as described in Theorem 1.

Challenge: Same as in Theorem 1.

Guess: A_{II} executes adaptive querying, and guesses b'. If A_{II} relays the tuples (x_3, y_3, h_2) to Query-H_2, it would know that $\hat{\sigma}$ is a flawed ciphertext. Then, C can output $abP = Z - d_v X$ as a program to solve the ECCDHP.

We define the following two cases:

π_1: C passing the query stage.

π_2: C passing the challenge stage.

We can deduce that:

$$\begin{aligned}
Pr[\pi_1] &= (1-\delta)^{q_{SK}}(1 - q_{USC}/2^l), \\
Pr[\pi_2|\pi_1] &= \delta, \\
Pr[C\ success] &= Pr[\pi_1 \wedge \pi_2] \\
&= Pr[\pi_1]Pr[\pi_2|\pi_1] \\
&= (1-\delta)^{q_{SK}} \cdot \delta(1 - \frac{q_{USC}}{2^l}) \\
&\geq \frac{(1 - \frac{q_{USC}}{2^l})}{eq_{SK}}
\end{aligned} \tag{22}$$

where $\delta = \frac{1}{q_{SK}+1}$.

Thus, C can be used with probability $\varepsilon' \geq \frac{(1-q_{USC}/2^l)}{eq_{SK}}$ to solve the ECCDHP if A_{II}'s advantage of success is ε. □

6.3. Unforgeability

Theorem 3. *If a Type I opponent A_I can successfully attack EUF-CLRSC-CMA2 for a non-negligible advantage ε in Game 3, then simulator C can solve the ECDLP with a probability $\varepsilon' \geq \varepsilon/[e(q_{SK} + q_{PSK} + q_{RPK})]$.*

Proof. Assume The simulator C receipts the tuple $(P, aP) \in G^2$. It computes the value of a in Game 3. Set $Pr(ID_i = ID^*) = \delta$.

Setup: The setup is the same as described in Theorem 1.

Query: The same rules as presented in Theorem 1.

Forgery: A_I returns a ciphertext $\sigma = \{C, \{s_i\}, X, L, I, W, TS\}$ that meets the requirements of Game 3. To forge another ciphertext σ^*, A_I replays queries Query-H_4 and Query-H_5 to gain another signcryption $\sigma^* = \{C^*, \{s_i^*\}, X^*, L, I^*, W^*, TS\}$. The intermediate values of the two signcryption are $(k, r_1, r_2, \ldots, r_n, e_1, e_2, \ldots, e_n)$ and $(k^*, r_1^*, r_2^*, \ldots, r_n^*, e_1^*, e_2^*, \ldots, e_n^*)$

in the correct order. Hence, when $i \in \{1, 2, \ldots, n\}$, the conditions $s_i \neq s_i^*$, $r_i \neq r_i^*$, and $e_i \neq e_i^*$ are established, so with the following calculation

$$s_s = [(1 + sk_s)^{-1}(k - r_s sk_s)] \tag{23}$$

$$s_s^* = [(1 + sk_s)^{-1}(k^* - r_s^* sk_s)] \tag{24}$$

$$sk_s = d_s + a \tag{25}$$

$$d_s = v_s + e_s x \tag{26}$$

Then, C computes $a = \frac{k^* - k - (s_s^* - s_s)}{s_s^* - s_s + r_s^* - r_s} - xe_s - v_s$.
We define three events as follows:
π_1: C adopts the Query stage.
π_2: $ID^* \in L$.
π_3: ID^* is the real signatory.
We can know that:

$$\begin{aligned} Pr[\pi_1] &= (1-\delta)^{q_{SK}+q_{PSK}+q_{RPK}} \\ Pr[\pi_2|\pi_1] &= \frac{n}{\delta} \\ Pr[\pi_3|\pi_2 \wedge \pi_1] &= \frac{1}{n} \\ Pr[C\ success] &= Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] \\ &= Pr[\pi_1] \cdot Pr[\pi_2|\pi_1] \cdot Pr[\pi_3|\pi_2 \wedge \pi_1] \\ &= (1-\delta)^{q_{SK}+q_{PSK}+q_{RPK}} \cdot \frac{n}{\delta} \cdot \frac{1}{n} \\ &\geq \frac{1}{e(q_{SK}+q_{PSK}+q_{RPK})} \end{aligned} \tag{27}$$

where $\delta = 1/(q_{SK} + q_{PSK} + q_{RPK} + 1)$.
We can deduce that $Pr[C\ success] \geq 1/[e(q_{SK} + q_{PSK} + q_{RPK})]$. □

Based on the forking lemma for ring signatures [26], C can solve the ECDLP for the probability of $\varepsilon' \geq \varepsilon/[e(q_{SK} + q_{PSK} + q_{RPK})]$, if the advantage A_{II} succeeds is ε.

Theorem 4. *If a Type II adversary A_{II} gains a notable advantage ε in Game 3, successfully compromising EUF-CLRSC-CMA2, it implies that a simulator C could potentially solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) with a probability $\varepsilon' \geq \varepsilon/(eq_{SK})$.*

Proof. Suppose the simulator C is provided with the tuple $(P, aP) \in G^2$. Its objective is to determine the value of a within Game 3. To achieve this, simulator C engages with the adversary A_I. Let us assume that ID^* represents the target identity, with $Pr(ID_i = ID^*) = \delta$ being the assigned probability.

Setup: The setup is the same as described in Theorem 2.

Query: The query phase follows the same rules as presented in Theorem 2.

Forgery: A_{II} returns a ciphertext $\sigma = \{C, \{s_i\}, X, L, I, W, TS\}$ that meets the requirements of Game 3. To forge another ciphertext σ^*, A_I replays queries Query-H_4 and Query-H_5 to gain another signcryption $\sigma^* = \{C^*, \{s_i^*\}, X^*, L, I^*, W^*, TS\}$. The intermediate values of the two signcryption are $(k, r_1, r_2, \ldots, r_n, e_1, e_2, \ldots, e_n)$ and $(k^*, r_1^*, r_2^*, \ldots, r_n^*, e_1^*, e_2^*, \ldots, e_n^*)$ in the correct order. Hence, when $i \in \{1, 2, \ldots, n\}$, the conditions $s_i \neq s_i^*$, $r_i \neq r_i^*$, and $e_i \neq e_i^*$ are established, so with the following calculation

$$s_s = [(1 + sk_s)^{-1}(k - r_s sk_s)] \tag{28}$$

$$s_s^* = [(1 + sk_s)^{-1}(k^* - r_s^* sk_s)] \tag{29}$$

$$sk_s = d_s + u_s \tag{30}$$

$$d_s = v_s + e_s a \tag{31}$$

thus $a = \frac{k^* - k - (s_s^* - s_s)}{e_s[(s_s^* - s_s + r_s^* - r_s) - u_s - v_s]}$.
We define three events as follows:
π_1: C adopts the Query stage.
π_2: $ID^* \in L$.
π_3: ID^* is the real signatory.
We can know that:

$$\begin{aligned} Pr[\pi_1] &= (1-\delta)^{q_{SK}} \\ Pr[\pi_2|\pi_1] &= \frac{n}{\delta} \\ Pr[\pi_3|\pi_2 \wedge \pi_1] &= \frac{1}{n} \\ Pr[C\ success] &= Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] \\ &= Pr[\pi_1] \cdot Pr[\pi_2|\pi_1] \cdot Pr[\pi_3|\pi_2 \wedge \pi_1] \\ &= (1-\delta)^{q_{SK}} \cdot \frac{n}{\delta} \cdot \frac{1}{n} \\ &\geq \frac{1}{eq_{SK}} \end{aligned} \tag{32}$$

where $\delta = 1/q_{SK}$. □

Drawing from the forking lemma to ring signatures (Ref. [26]), C has the ability to solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) with a probability $\varepsilon' \geq \varepsilon/(eq_{SK})$, given that the success rate of the advantage A_{II} is ε.

6.4. Anonymity

The collection L encompasses the public keys belonging to the legitimate senders within the ring. When validating the signcryption, the verifier applies a consistent formula using all public keys from L". Owing to the cryptographic attributes inherent in signcryption, the verifier cannot differentiate the true identity of the sender, thus preserving the anonymity of the original sender.

6.5. Traceability

When suspicious information is detected and there is a need to identify the true signer, TRA assesses the identity of the genuine signer using the I value within the suspicious signcrypt. Upon receiving the I value, TRA examines the ring public key set L to verify the identity ID_i of the true signer by validating the equation $k^{-1}I = PK_i + H_3(C, A)T_i$. The k^{-1} value in the equation is exclusively known to TRA, thus ensuring that conditional anonymity is preserved in the proposed CLRSC scheme.

6.6. Unlinkability

In the signcryption process, when generating the identifier I, the signer incorporates a variable β dependent on the message content ensuring that each message yields a unique I. Consequently, for different messages, the same signer calculates I differently. In the verification phase aimed at uncovering the true signer, the equation $k^{-1}I = PK_i + H_3(C, A)T_i$ is employed, with only TRA possessing the knowledge of the value k^{-1}. This ensures that only TRA has the capability to discern the identity of the actual signer.

6.7. Replay Attack Resistance

To prevent such situations, we incorporate timestamps into the encryption process, indicating the time of ciphertext transmission. If a ciphertext cannot be validated, indicating a potential replay attack, the insecure ciphertext will be discarded, and the sender will be notified to resend the ciphertext. Upon receiving a ciphertext, the verifier first examines whether the timestamp TS contained within the ciphertext satisfies the condition $|TS - TS_{cur}| \leq \triangle TS$, where TS_{cur} denotes the current timestamp, and $\triangle TS$ represents the maximum permissible time interval. If this condition is not met, ID_r rejects the ciphertext σ, thereby ensuring that intercepted and subsequently returned messages cannot pass the verification conducted by ID_r.

6.8. Anti-Malicious Gateway

As an edge computing node, ECN is likely to be a target for attackers. Being a semi-trusted gateway, ECN cannot guarantee that it will not be successfully attacked. Therefore, in our solution, ECN is designed not to have access to plaintext data, ensuring that information remains encrypted throughout the transmission process, thus reducing the risk of information leakage. In order to protect the message, after receiving it, ECN only partially decrypts it. ECN can only obtain Y through calculation, without CC's private key sk_v. As a result, it cannot obtain the message M, eliminating the possibility of message leaks.

6.9. Forward Security

When the system is compromised, measures are taken to prevent further escalation of losses. We designed the algorithm of Update-KeyPairs to regularly update the key. If a user accidentally loses the key, the security of the message before this cycle will not be questioned. Every time the user passes the previous cycle The private key and the random value u_i^t of this period are used to calculate the public-private key pair of this period. The key for each cycle is irregular, which prevents further damage due to key loss.

7. Performance Analysis

In this chapter, a comprehensive analysis of the scheme versus the existing alternatives is presented. The main tasks are as follows:

1. Functional analysis: the functionality of this paper is compared with classical papers, which are similar to existing schemes. The number of users in SGs is increasing rapidly and the complexity of the environment requires more functionality. Cryptographic parties with more functionalities are more in line with the developing SGs.
2. Computational efficiency analysis: in order to specifically analyze this scheme, a comparative analysis will be performed on ring signcryption with existing papers [3,23,27,28] and existing literature on aggregated signcryption papers [29–32], respectively. Suppose a ring has n members and m messages.
3. Communication cost analysis: In the comparison process, the communication cost is mainly reflected in two places: the communication cost of ECN and the communication cost of CC. A phase-by-phase comparison is made to show how the program can effectively solve a wider range of problems at a lower cost.

To ensure a fair comparison, we acquire the execution time of the most time-intensive operations by employing well-established encryption libraries such as pairing-based cryptography (PBC) and Miracl. We conduct simulations on a Lenovo Thinkpad laptop in China, featuring is Intel Core i5-9300H CPU and 16 GB RAM.

Comparing the scenarios at the same security level of 80 bits, for the scheme using the bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$, define the generating element of the additive group G_1 to be $\ddot{P}$ and the order to be $\ddot{q}$, and set the elliptic curve $\ddot{E} : y^2 = x^3 + x \; mod \; \ddot{p}$, $\ddot{p}$ and $\ddot{q}$ are numbers of size 64 bytes and 20 bytes. For the ECC-based scheme, we define the additive group G of order $\dot{q}$ on $\dot{E} : y^2 = x^3 + ax + b \; mod \; \dot{p}$, where $\dot{p}$ and $\dot{q}$ are two of size 20 bytes and $a, b \in Z_{\dot{p}}^*$.

7.1. Compare Algorithm Functions

In this section, we will select typical excellent papers [3,23,27–32] that are currently available and compare them. The main focus of attention is on the functionality of the programs and the problems they solve. The analyzed results are represented in Table 2.

Table 2. Comparison of Program Functions.

References	[3]	[23]	[27]	[28]	[29]	[30]	[31]	[32]	**Ours**
Traceability	✗	✗	✓	✓	✗	✓	✗	✗	✓
Confidentiality	✓	✓	✓	✓	✓	✓	✓	✓	✓
Anonymity	✓	✓	✓	✓	✓	✓	✓	✓	✓
Replay attack resistance	✓	✓	✓	✓	✗	✗	✗	✗	✓
Edge computing	✗	✗	✗	✗	✗	✗	✗	✗	✓
Reventing unsafe ECN	✗	✗	✗	✗	✗	✗	✗	✗	✓
User identity privacy protection	✓	✓	✓	✓	✗	✗	✗	✗	✓
Data privacy protection	✓	✗	✗	✗	✓	✓	✓	✓	✓
Update key	✗	✗	✗	✗	✗	✗	✗	✗	✓

From Table 2, we can clearly see that the scheme in this paper has more comprehensive functions and solves more problems at the same time. The primary distinction lies in this paper's utilization of edge computing to address the challenge of user surges in SG. In combination with edge computing At the same time, it also prevents the problem of malicious ECN. This is not available in the existing scheme. At the same time, we also added the function of updating the key to prevent the security of the previously sent information after someone's key is lost.

Through Table 2, it can be seen that the proposed scheme is more secure than the existing schemes. Literature [3,23,29,31,32] lack the tracking feature for malicious users, and literature [23,27,28] cannot perform data privacy protection. Except for this scheme, none of the existing typical schemes have forward and backward security.

The edge computing introduced in this scheme not only solves the problem of user proliferation but also resists attacks from malicious ECNs, which is a feature not available in existing schemes. In addition, this scheme adds a key update feature that protects the security of previously sent messages in case of key loss.

7.2. Computational Efficiency Analysis

In this experiment, we only focus on the more consuming operations, and operations that take less time are ignored, which does not affect the objectivity and fairness of our experiments. The notations corresponding to various computational operations are defined, and the corresponding elapsed times are listed in Table 3.

Table 3. Execution time of encryption operation.

Operation	**Operation**	**Executing Time (ms)**
T_h	hash-to-point operation	4.874
T_p	bilinear pairing operation	5.239
T_e	exponential operation	2.637
T_{Gm}	scale multiplication operation in G_1	2.896
T_m	point multiplication operation in G	1.156
T_a	point addition operation in G	0.023

In terms of computational cost, since the literature we compare are all certificateless schemes, the computational cost of this scheme in the key generation phase is not much different from the schemes we compare and is mostly $5T_m + T_a$. The computational costs in the ring signcryption phase, batch verification phase, aggregate signcryption phase, aggregate signcryption verification phase and tracking phase of this paper are $(n+5)T_m + nT_a$, $(n+2)T_m + nT_a$, $(n+2)T_m + nT_a$, $T_m + (m-1)T_a$, $(n+2)T_m + nT_a$ and $2T_m$, respectively.

In order to satisfy the fairness and reasonableness of the analysis, we analyze the computational efficiencies of schemes [3,23,27,28] and scheme [3,29–32] at different stages, respectively, and the analysis results are shown in Tables 4 and 5.

Table 4. Comparison of the computational efficiency of the ring signcryption part.

References	Ring Signcryption	Ring Verification	Batch Verification	Tracking
[3]	$(3n+2)T_m + (2n-1)T_a$	-	-	-
[27]	$(4n+3)T_m + (5n-3)T_a$	$(4n+2)T_m + (5n-1)T_a$	$(4n+2)mT_m + 5nT_a$	$3T_m + 3T_a$
[23]	$(2n+3)T_m$	$(n+2)T_m + nT_a$	-	-
[28]	$(n+3)T_{Gm} + T_p + (2n-2)T_a$	$nT_{Gm} + 4T_p$	-	$2T_{Gm} + 2T_p$
Ours	$(n+5)T_m + nT_a$	$(n+2)T_m + nT_a$	$(n+2)T_m + nT_a$	$2T_m$

Table 5. Comparison of the computational efficiency of the aggregate signcryption part.

References	Aggregate Signcryption	Aggregate Verification
[3]	$(m-1)T_a$	$(m+1)T_m + (m-1)T_a$
[29]	$(m-1)T_a$	$2mT_{Gm} + T_p$
[30]	$(2m+2)T_m + (2m+2)T_a$	$(m+3)T_m$
[31]	$(m-1)T_a$	$3mT_{Gm} + mT_p + 3mT_a$
[32]	$(m-1)T_a$	$(2m+1)T_m + (3m-1)T_a$
Ours	$T_m + (m-1)T_a$	$(n+2)T_m + nT_a$

It can be seen from Table 4 that our scheme has a significant improvement compared with the existing schemes in the ring signcryption stage. From Table 5, we can see that the scheme in this paper adds a multiplication operation T_m to the existing advanced schemes in the aggregation process, but it has a huge advantage in the verification process.

In order to more realistically simulate the application scenarios of the smart grid, we assume that $n = 10$, $m = 100$, simulate the computational efficiency of each scheme and show the experimental results in Tables 6 and 7. In order to show the difference between the various schemes more clearly, we also draw Figure 3 and 4 based on Tables 6 and 7.

In Figure 3, the batch verification part of paper [27] is too inefficient. In order to make the results in Figure 3 more obvious, we reduced this value from 4856.35 to 100, and the real data are larger than those shown in Figure 3. Similarly, in Figure 4, the aggregated signcryption part of the paper [30] and the aggregate verification part of the papers [29,31,32] are too large. In order to make the picture show the difference between each scheme, we uniformly reduce these data to 150.

Table 6. Comparison of the computational efficiency of the ring signcryption part.

References	Ring Signcryption (bytes)	Ring Verification (bytes)	Batch Verification (bytes)	Tracking (bytes)
[3]	37.429	-	-	-
[27]	50.789	49.679	4856.35	3.537
[23]	26.588	14.102	-	-
[28]	43.301	49.916	-	24.405
Ours	17.57	14.102	14.102	2.312

Table 7. Comparison of the computational efficiency of the aggregate signcryption part.

References	Aggregate Signcryption (bytes)	Aggregate Verification (bytes)
[3]	2.277	119.033
[29]	2.277	150
[30]	150	119.068
[31]	2.277	150
[32]	2.277	150
Ours	2.433	14.102

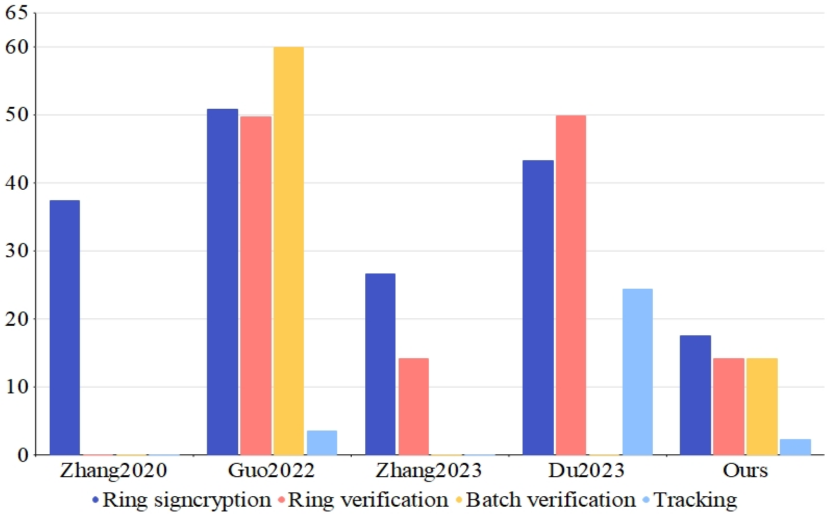

Figure 3. Efficiency comparison of ring signcryption stage (n = 10, m = 100) [3,23,27,28].

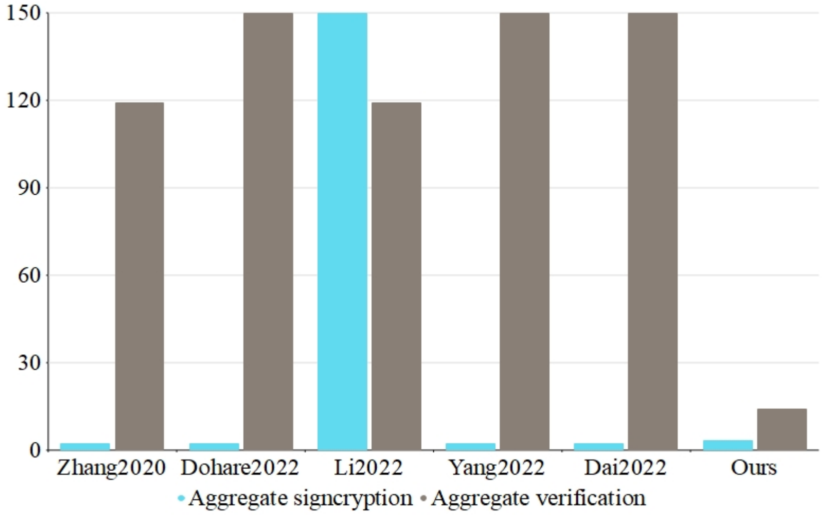

Figure 4. Efficiency comparison of aggregation signcryption stage (n = 10, m = 100) [3,29–32].

We divide the comparison process into two parts: the ring signcryption part and the aggregation signcryption part. Tables 4 and 6 and Figure 3 show the ring signcryption part, and Tables 5 and 7 and Figure 4 show the aggregation signcryption part. The analysis shows that in terms of computational efficiency, this solution is more efficient than existing schemes in both ring signcryption and aggregation signcryption.

In the ring signcryption phase, ref. [3] does not verify the signcryption, resulting in lower efficiency, and lacks verification, batch verification, and ciphertext tracing algorithms. Ref. [27], while having a complete algorithm, suffers from lower efficiency, taking twice the time compared to our approach. Ref. [23] shows slightly lower efficiency and lacks a batch verification algorithm, making it unsuitable for practical scenarios. Ref. [28] exhibits not only lower efficiency but also a longer time for the tracing algorithm, and lacks a batch verification algorithm. Therefore, in the ring signcryption phase, our approach demonstrates significantly higher efficiency compared to existing works, with a complete algorithm.

In the aggregate signcryption phase, although the signcryption part of our approach is slightly increased compared to [3,29,31,32], the overall efficiency is better than existing works. There is a noticeable improvement in the efficiency of aggregate ciphertext verification. It is evident that our approach demonstrates better efficiency in the aggregate signcryption phase compared to existing works overall.

The analysis shows that in terms of computational efficiency, this solution is more efficient than existing schemes in both ring signcryption and aggregation signcryption.

7.3. Communication Cost Analysis

In the model of this paper, the ciphertext is transmitted in two main places. User ID_s transmits the ciphertext to ID_r after signcryption of the message. ID_r performs aggregation and sends the aggregated ciphertext to ID_v. We compare the two phases separately in Tables 8 and 9.

Table 8. Comparison of the communication cost analysis of the ring signcryption part.

References	Ring Signcryption Communication Cost (bytes)
[3]	$(n+3)\|G\| + \|Z_{\dot{q}}^*\| + \|M\| + \|TS\| + \|L\| = 708$
[27]	$3\|G\| + (n+1)\|Z_{\dot{q}}^*\| + \|M\| + \|TS\| + \|L\| = 772$
[28]	$(n+2)\|G_1\| + \|M\| + \|TS\| + \|L\| = 1704$
[23]	$(n+1)\|G\| + 3\|Z_{\dot{q}}^*\| + \|M\| + \|TS\| + \|L\| = 608$
Ours	$3\|G\| + (n+1)\|Z_{\dot{q}}^*\| + \|M\| + \|TS\| + \|L\| = 772$

Table 9. Comparison of the communication cost analysis of the aggregate signcryption part.

References	Aggregate Signcryption Communication Cost (bytes)
[3]	$(n+3)\|G\| + \|Z_{\dot{q}}^*\| + m\|M\| + \|TS\| + \|L\| = 16,548$
[29]	$(m+1)\|G_1\| + \|Z_{\dot{q}}^*\| + m\|M\| = 28,948$
[30]	$(m+1)\|G_1\| + 4m\|M\| = 68,040$
[31]	$(m+1)\|G\| + m\|M\| = 28,800$
[32]	$m\|G\| + m\|M\| = 28,820$
Ours	$(m+2)\|G\| + n\|Z_{\dot{q}}^*\| + m\|M\| + \|TS\| + \|L\| = 29,084$

According to the experimental parameters we wrote at the beginning of this chapter, we can obtain $|Z_{\dot{q}}^*| = |Z_{\ddot{q}}^*| = 20$ bytes and set $n = 10$, $m = 100$ similar to the above section. By simple calculation, we can obtain that the length of message $|M| = l = 160$ bytes, the length of the timestamp is $|TS| = 4$ bytes and the public key set $|L| = 4$ bytes, the elements in G, G_1 and G_2 are 40 bytes and 128 bytes, and are signified as $|G|$ and $|G_1|$, respectively.

To make a clearer comparison of the communication costs, we set the number of ring members at 5 to better show the differences between the schemes. Based on Table 5, we can calculate the following. The costs of the four ring signcryption schemes and the scheme in this paper are 708, 772, 1704, 608 and 772, respectively. In this link, this paper is obviously better than the existing similar schemes. Combined with Table 8, we know that although our cost is not the lowest, it is not much different from the cost of the current excellent solutions.

After analysis, in terms of communication cost, the present scheme is lower than the literature [28] in the ring signing secret phase, and the same as the literature [27], which is not the lowest but still the lowest cost among the existing typical schemes. In the aggregate signcryption phase, the communication cost of the present scheme is lower than that of the literature [23], and not much different from that of the literature [29,31,32]. Taken together, this scheme is not the best in terms of communication cost, but the increase is less compared to typical schemes. It is worthwhile to sacrifice a small amount of communication cost to add more security and higher efficiency.

8. Conclusions

Protecting user privacy in SG is critical to its development. However, none of the existing solutions are suitable for SG, or cannot better solve the existing problems. This is very unfavorable for the development of SG. In this paper, we propose a certificateless aggregated ring signcryption scheme with conditional privacy in SG. By incorporating aggregate signcryption to improve computational efficiency, utilizing timestamps to counter replay attacks, and employing multi-layer encryption to resist malicious gateways, security has been enhanced. Through security analysis, it is proved that the scheme can resist external attacks and internal malicious KGC threats and has more comprehensive functions. Through the efficiency analysis experiment, it can be seen that compared with the existing schemes with the same function, our scheme does not require bilinear pairing and is faster. Ring signcryption and aggregate signcryption are performed under the same structure, improving computational efficiency and communication costs, which have obvious advantages over existing schemes.

Author Contributions: Conceptualization, H.S.; methodology, L.Z.; formal analysis, H.G.; data curation, Z.L.; supervision, S.L.; funding acquisition, T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (No. 62102311), in part by the Key Research and Development Program of Shaanxi (No. 2024GX-YBXM-571), and in part by the Doctoral Fund of Henan Polytechnic University (No. SKB2022-12).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tuballa, M.L.; Abundo, M.L. A review of the development of Smart Grid technologies. *Renew. Sustain. Energy Rev.* **2016**, *59*, 710–725. [CrossRef]
2. Varghese, B.; Wang, N.; Barbhuiya, S.; Kilpatrick, P.; Nikolopoulos, D.S. Challenges and opportunities in edge computing. In Proceedings of the IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, USA, 18–20 November 2016; pp. 20–26.
3. Zhang, S.; Rong, J.; Wang, B. A privacy protection scheme of smart meter for decentralized smart home environment based on consortium blockchain. *Int. J. Electr. Power Energy Syst.* **2020**, *121*, 106140. [CrossRef]
4. Cai, Y.; Zhang, H.; Fang, Y. A conditional privacy protection scheme based on ring signcryption for vehicular ad hoc networks. *IEEE Internet Things J.* **2020**, *8*, 647–656. [CrossRef]
5. Li, F.; Luo, B.; Liu, P. Secure information aggregation for smart grids using homomorphic encryption. In Proceedings of the first IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, USA, 4–6 October 2010; pp. 327–332.
6. Singh, P.; Masud, M.; Hossain, M.S.; Kaur, A. Blockchain and homomorphic encryption-based privacy-preserving data aggregation model in smart grid. *Comput. Electr. Eng.* **2021**, *93*, 107209. [CrossRef]
7. Feng, C.; Wang, Y.; Chen, Q.; Ding, Y.; Strbac, G.; Kang, C. Smart grid encounters edge computing: Opportunities and applications. *Adv. Appl. Energy* **2021**, *1*, 100006. [CrossRef]
8. Metke, A.R.; Ekl, R.L. Security technology for smart grid networks. *IEEE Trans. Smart Grid* **2010**, *1*, 99–107. [CrossRef]
9. McDaniel, P.; McLaughlin, S. Security and privacy challenges in the smart grid. *IEEE Secur. Priv.* **2009**, *7*, 75–77. [CrossRef]
10. Depuru, S.S.S.R.; Wang, L.; Devabhaktuni, V.; Gudi, N. Smart meters for power grid—Challenges, issues, advantages and status. In Proceedings of the 2011 IEEE/PES Power Systems Conference and Exposition, Phoenix, AZ, USA, 20–23 March 2011; pp. 1–7.
11. Liu, J.; Xiao, Y.; Li, S.; Liang, W.; Chen, C.L.P. Cyber security and privacy issues in smart grids. *IEEE Commun. Surv. Tutorials* **2012**, *14*, 981–997. [CrossRef]
12. Li, D.; Yang, Q.; Yu, W.; An, D.; Zhang, Y.; Zhao, W. Towards differential privacy-based online double auction for smart grid. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 971–986. [CrossRef]

13. Tian, X.; Song, Q.; Tian, F. Multidimensional data aggregation scheme for smart grid with differential privacy. *Int. J. Netw. Secur.* **2018**, *20*, 1137–1148.
14. Zheng, Z.; Wang, T.; Bashir, A.K.; Alazab, M.; Mumtaz, S.; Wang, X. A decentralized mechanism based on differential privacy for privacy-preserving computation in smart grid. *IEEE Trans. Comput.* **2021**, *71*, 2915–2926. [CrossRef]
15. Rivest, R.L.; Shamir, A.; Tauman, Y. How to leak a secret. In *Advances in Cryptology—ASIACRYPT 2001: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, QLD, Australia, 9–13 December 2001*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 552–565.
16. Han, W.; Xiao, Y. Privacy preservation for v2g networks in smart grid: A survey. *Comput. Commun.* **2016**, *91*, 17–28. [CrossRef]
17. Wang, Q.; Chen, J.; Zhuang, L. Batch verification of linkable ring signature in smart grid. In *Frontiers in Cyber Security, Proceedings of the International Conference on Frontiers in Cyber Security, Xi'an, China, 15–17 November 2019*; Shen, B., Wang, B., Han, J., Yu, Y., Eds.; Springer: Singapore, 2019; Volume 1105, pp. 161–176.
18. Tang, F.; Pang, J.; Cheng, K.; Gong, Q. Multiauthority traceable ring signature scheme for smart grid based on blockchain. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5566430. [CrossRef]
19. Liu, Y.; He, D.; Bao, Z.; Wang, H.; Khan, M.K.; Choo, K.R. An efficient multilayered linkable ring signature scheme with logarithmic size for anonymous payment in vehicle-to-grid networks. *IEEE Trans. Intell. Veh.* **2022**, *8*, 2998–3011. [CrossRef]
20. Liu, S.; Liu, Z.; Liang, J.; Zhang, W.; Heng, Z. A secure certificateless ring signcryption scheme based on SM2 algorithm in smart grid. *Comput. Commun.* **2024**, *218*, 188–197. [CrossRef]
21. Zhang, S.; Zheng, T.; Wang, B. A privacy protection scheme for smart meter that can verify terminal's trustworthiness. *Int. J. Electr. Power Energy Syst.* **2019**, *108*, 117–124. [CrossRef]
22. Wang, H.; Wang, L.; Wen, M.; Chen, K.; Luo, Y. A lightweight certificateless aggregate ring signature scheme for privacy-preserving in smart grids. *Wirel. Pers. Commun.* **2022**, *126*, 1577–1599. [CrossRef]
23. Zhang, S.; Guo, Y.; Wang, B. A privacy protection scheme for bidding users of peer-to-peer electricity call auction trading in microgrids. *IEEE Syst. J.* **2023**, *17*, 3316–3327. [CrossRef]
24. *GM/T 0003.1–2012*; Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves. National Standards of the People's Republic of China: Beijing, China, 2012.
25. Teng, D.; Yao, Y.; Wang, Y.; Zhou, L.; Huang, C. An sm2-based traceable ring signature scheme for smart grid privacy protection. In Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications, Dalian, China, 24–26 November 2022; pp. 296–313.
26. Herranz, J.; Sáez, G. Forking lemmas for ring signature schemes. In Proceedings of the International Conference on Cryptology in India, New Delhi, India, 8–10 December 2003; pp. 266–279.
27. Guo, R.; Xu, L.; Li, X.; Zhang, Y.; Li, X. An efficient certificateless ring signcryption scheme with conditional privacy-preserving in vanets. *J. Syst. Archit.* **2022**, *129*, 102633. [CrossRef]
28. Du, H.; Wen, Q.; Zhang, S.; Gao, M. An improved conditional privacy protection scheme based on ring signcryption for vanets. *IEEE Internet Things J.* **2023**, *10*, 17881–17892. [CrossRef]
29. Dohare, I.; Singh, K.; Ahmadian, A.; Mohan, S. Certificateless aggregated signcryption scheme (class) for cloud-fog centric industry 4.0. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6349–6357. [CrossRef]
30. Li, K.; Shi, R.; Wu, M.; Li, Y.; Zhang, X. A novel privacy-preserving multi-level aggregate signcryption and query scheme for smart grid via mobile fog computing. *J. Inf. Secur. Appl.* **2022**, *67*, 103214. [CrossRef]
31. Yang, Y.; He, D.; Vijayakumar, P.; Gupta, B.B.; Xie, Q. An efficient identity-based aggregate signcryption scheme with blockchain for iot-enabled maritime transportation system. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 1520–1531. [CrossRef]
32. Dai, C.; Xu, Z. Pairing-free certificateless aggregate signcryption scheme for vehicular sensor networks. *IEEE Internet Things J.* **2022**, *10*, 5063–5072. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.