Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 20(11): 20295–20316.
DOI: 10.3934/mbe.2023898
Received: 02 August 2023
Revised: 26 September 2023
Accepted: 12 October 2023
Published: 08 November 2023

Research article

A heterogeneous signcryption scheme for smart grid with trusted multi-ciphertext equality test

Xiaodong Yang[1], Ruixia Liu[1,*], Bin Shu[2], Ningning Ren[1] and Wenjia Wang[1]

[1] College of Computer Science and Engineering, Northwest Normal University, Lanzhou, China
[2] China Telecom Wanwei Information Technology Company Limited, Lanzhou, China

* **Correspondence:** Email: 2021222235@nwnu.edu.cn; Tel: +8615835144606.

Abstract: Energy utilization rates have been largely improved thanks to the wide application of smart grids, thereby realizing the reliable, economic and efficient operation of the grids. However, such an application is also accompanied by many security issues. In response to the many problems within existing security schemes, such as not supporting the communication between heterogeneous cryptosystems, low security levels and a low data retrieval efficiency, a heterogeneous signcryption (HSC) scheme that supports a trusted multi-ciphertext equality test (MET) is proposed. The adoption of the HSC helps to identify secure communications from identity-based cryptosystems to certificateless cryptosystem, eliminates the certificate management problems in the traditional public key cryptography scheme, and ensures the confidentiality and authentication of power data. The introduction of the MET technology can avoid the high cost of equality test calculations after grouping ciphertexts in pairs. Using blockchain and smart contract technologies ensure the credibility of test results and eliminates the reliance on trusted cloud servers. Under the random oracle model, on the basis of the bilinear Diffie-Hellman, the computational Diffie-Hellman and the q-strong Diffie-Hellman problems, this paper proves that the scheme proposed herein meets the requirements of indistinguishability and one-way security under adaptive choice ciphertext attacks, and the unforgeability under the adaptive choice message attack. From the findings of the analysis, it has been shown that the proposed scheme satisfies more security attributes and requires lower computational overhead compared to similar schemes.

Keywords: smart grid; multi-ciphertext equality test; heterogeneous signcryption; blockchain

1. Introduction

A smart grid [1–3] is a new type of modern grid combining such technologies as advanced sensing measurements, information communications, and an automatic control, and is highly integrated within the grid infrastructure. Through the two-way transmission of real-time monitoring and control data, it can balance the power generation and demand of the entire power grid and is featured by a strong stability and self-healing ability. The components and equipment of a smart grid are usually distributed in a large geographical area, covering the whole process from power generation to users, thereby making it a complex heterogeneous network [4]. As the size of smart grid data rapidly increases, the storage and processing of big data has become a top priority. This big problem has been solved with the emergence of cloud computing technology [5]. Having powerful data processing capabilities, the cloud can provide users with computing services without being limited by time and location. In the cloud-assisted smart grid, numerous users' power data are outsourced to the cloud server (CS) for processing, thus alleviating the problems of high storage and computing overhead faced by the entities of smart grids. However, since the CS is not trusted, in order to ensure the confidentiality of the user's power data, data is often stored in the cloud in the form of ciphertext, thereby making data retrieval difficult and greatly reducing data availability.

To solve the conflict between data confidentiality and efficient retrieval, Boneh et al. [6] proposed a public key encryption scheme with a keyword search (PKE-KS). The proposed scheme is capable of retrieving target data with keywords from the CS without decryption, and further capable of downloading and decrypting on a similar basis. However, this scheme only supports the retrieval of ciphertexts encrypted with the same public key, thereby indicating certain limitations. To break this limitation, Yang et al. [7] first proposed a public key encryption scheme (PKE-ET) that supports an equality test (ET). It allows users to compare two ciphertexts encrypted with different public keys to determine whether their corresponding underlying plaintexts are the same. Since then, many PKE-ET schemes have been proposed [8–10]. However, these schemes only support ET after grouping ciphertexts in pairs. In environments with multiple users and ciphertexts, their retrieval efficiency is low and the computational overhead is high. To address this problem, Susilo et al. [11] proposed an PKE-MET supporting multi-ciphertext equality test (MET). This scheme supports utilizing the CS to test the equality of multiple ciphertexts at the same time, thus greatly reducing the computational overhead and improving the efficiency of the ciphertext retrieval.

Although the schemes [7–11] ensure the confidentiality of data through encryption, it does not achieve data authentication. In the smart grid, if the user's power data is not verifiable, it may not only cause data users (DUs) such as the distribution substation to waste computing resources for useless processing, but it may also lead to the incorrect judgment of the users' power consumption, resulting in unstable distribution and power failures.

In order to meet the requirements of data confidentiality and authentication at the same time, many scholars at home and abroad have combined ET with signcryption technology [12] and proposed signcryption schemes supporting ET [13–15]. However, their schemes only support the communication between single cryptosystems. A smart grid is a complex heterogeneous network, thereby making these schemes not applicable. To better apply the ET schemes to heterogeneous systems, Xiong et al. [16] proposed a heterogeneous signcryption (HSC) scheme supporting ET for the Industrial Internet of Things (IIOT), thus allowing sensors in the public key infrastructure (PKI) to execute data encryption and upload ciphertext to CS. When users in the identity-based cryptosystem (IBC) want to search and download data, the scheme entrusts CS to execute ET on ciphertext. Shan and Zhuang [17] utilized a game-theoretic approach to model attacks and the defenses of smart grids. This study provides valuable insights into the analysis of security strategies in small grid systems. However, in the test phase, the ciphertexts were grouped in pairs and compared one by one. Therefore, it requires a high computational overhead and is not suitable for multi-user scenarios.

In the application scenario of the smart grid, we propose an HSC scheme supporting MET. The main contributions are as follows.

1) The adoption of the HSC enables communication from IBC to a certificateless cryptosystem (CLC), eliminates the certificate management problem in the traditional public key cryptography scheme, and ensures the confidentiality, integrity and authentication of the user's power data in the smart grid.
2) The introduction of the MET allows testers to execute ET on multiple ciphertexts at the same time according to the user's trapdoors, thus realizing the safe and efficient retrieval of ciphertexts. It avoids the problems of a high computational overhead in traditional schemes supporting ciphertext ET, which divides ciphertexts into pairs before the ET.
3) Using blockchain technology, the MET operation is performed by a smart contract deployed on the alliance chain platform, resulting in the test operation's decentralization.
4) In the random oracle model (ROM), based on the bilinear Diffie Hellman (BDHP), the computational Diffie-Hellman (CDHP) and the q-strong Diffie-Hellman (q-SDHP) problems, the confidentiality, one-way security and unforgeability of the proposed scheme have been proven.
5) We compared the proposed scheme with several similar schemes in terms of functions, security attributes and computational overhead. The analysis findings have shown that our scheme has more functions, higher security attributes and a lower computational overhead.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes the preliminaries. In Section 4, we provide the system design. In Section 5, the algorithm processes of our scheme are presented. The correctness analyses, security, and efficiency of the proposed scheme are discussed in Sections 6–8, respectfully. Finally, Section 9 summarizes this paper.

2. Related works

In 2010, Yang et al. [7] first proposed the PKE-ET scheme, thereby allowing anyone to determine whether the underlying plaintext corresponding to two ciphertexts encrypted with different public keys was equal by executing an ET. However, this scheme failed to meet the requirement of indistinguishability under plaintext attacks. To solve this problem, Tang [18] introduced the fine-grained authorization mechanism into the ET scheme and proposed the fine-grained authorisation PKEET scheme (FG-PKEET). In this scheme, after the user negotiated to generate a test trapdoor, it would be provided to the agent to execute the ET. This scheme improves the security of the user's data but increased the storage cost of the agent. Since then, Tang has proposed the PKEET scheme [19] with authorization of different granularity (ADG-PKEET) and the all-or-nothing PKEET scheme (AON-PKEET) [20]. Inspired by the work of Tang, many PKE-ET schemes supporting authorization were proposed [21–23].

In order to improve the security and usability of the PKE-ET scheme, many scholars proposed to combine PKE-ET with various cryptosystems. By combining identity-based encryption (IBE) with PKE-ET, Ma [24] proposed the IBE-ET scheme, which used the identity of the receiver for encryption, thus avoiding the problems in certificate management. Qu et al. [25] integrated certificateless public key encryption into the PKE-ET scheme to avoid the key escrow problem in IBE-ET. Xiong et al. [13] proposed an ET scheme for the Internet of Vehicles (IoV) by combining identity based signcryption, thus realizing the classification of vehicle messages and ensuring the confidentiality and authentication of vehicle data. Yang et al. [15] proposed a certificateless ciphertext ET signcryption scheme for wireless body area networks by combining blockchain technology, eliminating the dependence of ET operations on trusted CS. Xiong et al. [26] proposed an HSC scheme to support ET by combining HSC, thus realizing the ciphertext data retrieval between IBC and PKI heterogeneous systems. Hou et al. [27] proposed an HSC scheme from PKI to CLC that supports ET. Zhao et al. [28] designed a certificateless signcryption scheme with ciphertext ET, and defined its framework and security model.

To solve the high-cost problem of the ET calculation after dividing the ciphertexts into pairs, Susilo et al. [11] proposed an MET public key encryption scheme, which realized the simultaneous retrieval of multiple ciphertexts by CS and reduced the calculation cost. However, the scheme failed to meet the requirements of authentication, thus indicating a low level of security. Furthermore, in the schemes described above, the test operation is carried out by a semi-trusted CS. If the CS either fails unexpectedly or is maliciously attacked and produces incorrect results, it may result in an incorrect diagnosis of the user's power consumption and even leak the user's privacy. Yang et al. [29] proposed an aggregated signcryption scheme for wireless body area networks that supports multi-ciphertext equivalence tests. The scheme aggregates multiple ciphertexts for signcryption, thereby leading to reduced computational overhead.

3. Preliminaries

3.1. Cramer's rule

If the coefficient determinant corresponds to the non-homogeneous linear equation group Eq (1), where n unknowns x_i satisfies $\det(V) \neq 0$, it has a unique solution.

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n = b_2 \\ \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n = b_n \end{cases} \tag{1}$$

3.2. Vandermonde determinant

Equation (2) is a Vandermonde matrix. The corresponding Vandermonde has computational properties $\det(V) = \prod_{1 \le i < j \le n} (a_i - a_j)$.

$$V = \begin{bmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^{n-1} \\ 1 & a_2 & a_2^2 & \cdots & a_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_n & a_n^2 & \cdots & a_n^{n-1} \end{bmatrix} \tag{2}$$

3.3. Bilinear pairing

Let q be a large prime number. G_1 and G_2 are additive cyclic and multiplicative cyclic groups of order q, respectively. P is the generator. $e: G_1 \times G_1 \to G_2$ is a bilinear map that satisfies the following three conditions:

1) Non-degeneracy: $e(P,P) \neq 1$;
2) Bilinearity: For any $a,b \in \mathrm{Z}_q^*$, $e(aP,bP) = e(P,P)^{ab}$;
3) Computability: For any $E,Q \in G_1$, there is an algorithm that can calculate $e(E,Q)$.

3.4. Computational assumptions

G_1 and G_2 are additive and multiplicative cyclic groups of prime q, respectively. P is the generator. The relevant and difficult problems are defined below:

1) BDHP: Given (P,aP,bP,cP) where $a,b,c \in Z_q^*$ and a bilinear map $e: G_1 \times G_1 \to G_2$, it is hard to calculate $e(P,P)^{abc}$;
2) CDHP: Given (P,aP,bP), in which $a,b \in Z_q^*$, it is hard to calculate abP;
3) q-SDHP: Given $q+1$ vectors $(P,aP,\cdots,a^qP)$ where $a,q \in Z_q^*$, it is hard to calculate $(k,\frac{1}{k+a}P)$.

3.5. Blockchain and smart contracts

Blockchains [30] can eliminate the dependence on trusted clouds and realize the decentralization of cryptographic schemes. Blockchains are divided into public chains, alliance chains and private chains according to the degree of openness. The public chain is a consensus blockchain open to everyone, which allows everyone to issue and confirm transactions. The alliance chain is a blockchain jointly built and maintained by multiple organizations. Each participant builds a consensus mechanism through a contract to achieve partial decentralization. Compared with the public chain, it has a stronger controllability. The trust degree of the private chain is lower than that of the public chain, and its write and read permissions are controlled by a centralized organization, which has a higher flexibility. The blockchain uses a specific reward mechanism to motivate each node to provide either computing power or resources and uses cryptographic algorithms to ensure the security of transactions.

The smart contract [15] is a piece of code deployed on the blockchain platform, which can conduct trusted transactions without relying on third parties. Blockchain nodes deploy predefined rules on the blockchain in the form of smart contracts by publishing transactions. When a transaction meets the trigger conditions, the smart contract will automatically perform relevant calculations and record the transaction information in the block in the chain. Smart contracts provide the blockchain application layer with various interfaces responsible for either storing or processing external data, thereby enabling blockchain to replace semi-trusted CSs to perform operations such as ciphertext retrieval. Using blockchain and smart contract technology eliminates the reliance on trusted CSs, improves the system scalability, ensures better maintainability and upgradability of the system, and minimizes the adverse effects of these updates on the power grid operation.

4. System design

4.1. Security goal

The scheme needs to achieve security under the following four kinds of adversaries:

1) Type-I: Those who can't obtain the system master key and user's trapdoor, but can replace the public key of any user;

2) Type-II: Those who can obtain the system master key, but cannot obtain the user's trapdoor, nor can they replace the user's public key;

3) Type-III: Those who can't obtain the system master key, but can obtain the user's trapdoor and replace the public key of any user;

4) Type-IV: Those who can obtain the system master key and user's trapdoor, but cannot replace the user's public key.

Facing the four types of adversaries, our scheme can achieve the following security goals.

1) Integrity and confidentiality of power data

The smart grid transmits and stores the power data of numerous users. The leakage and illegal tampering of power data will not only bring economic losses, but also endanger the users' personal safety. Our scheme achieves the integrity and confidentiality of type-I and type-II adversary attacks in the transmission of power data through HSC technology.

2) Unforgeability of user's signature

The receivers and illegal users may forge the user's electricity data to obtain illegal benefits. To avoid this problem, our scheme adopts HSC technology, thus ensuring the unforgeability of power data.

3) One-way security of trapdoor

In the process of MET, plaintext messages related to ciphertexts cannot be obtained through trapdoors. In the test process, the proposed scheme guarantees the one-way security of the trapdoor against adversary Type-III and Type-IV.

4) The credibility of the test result

The MET phase can determine whether the plaintexts of ciphertexts are equal without unsigncryption. It is necessary to output a correct result during the test phase.

4.2. System model

The system model of the proposed scheme is shown in Figure 1. It consists of six entities: key generation center (KGC), private key generation center (PKG), CS, data owner (DO), DU and blockchain. The functions of each entity are described below.

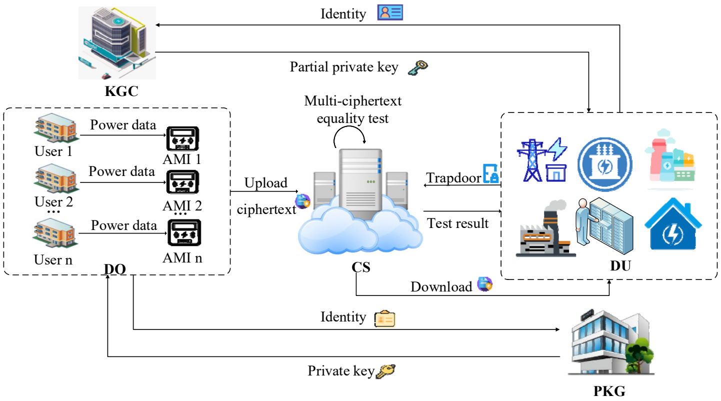

Figure 1. System model.

1) KGC: It generates the master key, the system parameter, and a partial private key for DUs in the CLC cryptosystem.

2) PKG: It generates the master key and system parameters, and issues the private key to the DU in the IBC cryptosystem.

3) DO: It is usually an intelligent terminal device, such as an advanced metering infrastructure (AMI), which is responsible for collecting the users' power data and uploading it to the CS after signcrypting.

4) CS: It stores the power data ciphertexts uploaded by the DO.

5) Blockchain: A consortium chain with smart contracts is deployed. It is responsible for downloading power consumption ciphertext data from CS, performing tests and returning the test results.

6) DU: This represents the power companies. It uploads a trapdoor and requests a blockchain for MET, downloads and unsigncrypts ciphertexts of the power data.

5. Scheme design

A) Setup

Given the system safety parameter λ, KGC selects the large prime number q and the cyclic groups G_1 and G_2 of the order q. P is the generator.

1) KGC defines a bilinear map $e: G_1 \times G_1 \rightarrow G_2$, calculates $g = e(P, P)$, and chooses hash functions $H_1 : \{0,1\}^* \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \rightarrow G_1$, $H_3 : G_1 \rightarrow \{0,1\}^{l_m + l_0}$, $H_4 : \{0,1\}^* \rightarrow \{0,1\}^{\lambda}$ and $H_5 : G_2 \rightarrow \{0,1\}^{l_m + l_0}$, in which l_m represents the length of the plaintext and l_0 represents the length of $|Z_q^*|$.

2) PKG randomly selects the master key $s_1 \in Z_q^*$ and calculates the public key $Pub_1 = s_1 P$.

3) KGC randomly selects the master key $s_2 \in Z_q^*$ and calculates $Pub_2 = s_2 P$. The output $params = \{\lambda, q, P, G_1, G_2, e, g, Pub_1, Pub_2, H_1, H_2, H_3, H_4, H_5\}$.

B) IBC-Gen

Key-extract: The DO sends the identity ID_s to PKG. The PKG calculates $SK_s = \dfrac{1}{H_1(ID_s) + s_1} P$ as private key and returns it safely to the DO.

C) CLC-Gen

1) PPK-Gen: The DU sends the identity ID_r to the KGC. It calculates a partial private key $D_r = s_2 H_2(ID_r)$ and returns to the DU safely.

2) SV-Set: The DU randomly selects the secret value $x_r \in Z_q^*$.

3) Key-Gen: The DU calculates $PK_r = x_r P$ and $SK_r = (D_r, x_r)$.

D) Trapdoor

Input the user's private key SK_r and output the trapdoor $td_r = SK_{r1}$.

E) Signcryption

Given the system parameter *params*, the plaintext m, the identity of DU, ID_r and public key PK_r, the DO executes the following operations:

1) Calculate $f_0 = H_1(m \| n)$, $f_1 = H_1(m \| n \| f_0)$, $\cdots$, $f_{n-1} = H_1(m \| n \| f_0 \| \cdots \| f_{n-2})$, and let $f(x) = f_0 + f_1 x + \cdots + f_{n-1} x^{n-1}$, in which n is the number of ciphertext that allows the MET;

2) Randomly select $r_1, r_2, X \in Z_q^*$; and calculate $U_1 = r_1 PK_r$, $Q_r = H_2(ID_r)$ and $U_2 = e(Pub_2, Q_r)^{r_2}$;

3) Calculate $C_1 = r_1 P$, $C_2 = r_2 P$, $C_3 = (m \| r_1) \oplus H_3(U_1) \oplus H_5(U_2)$, $C_4 = (r_1 + H_1(m)) SK_s$, $C_5 = (X \| f(X)) \oplus H_3(U_2)$ and $C_6 = H_4(n \| C_1 \| \cdots \| C_5 \| U_2 \| f_0 \| \cdots \| f_{n-1})$;

4) Upload $\delta{=}(C_1, C_2, C_3, C_4, C_5, C_6, n)$ to CS.

F) MET

Data users send trapdoors $td_i, i \in \{1, 2, \cdots, t\}$ to the blockchain, which executes the following MET operations to t ciphertexts $\delta_i{=}(C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}, C_{i,5}, C_{i,6}, n_i)$:

1) Test whether $n_1 = n_2 = \cdots = n_t = t$ is true; if yes, execute the following calculation. Otherwise, return the error symbol $\perp$;

2) Calculate $U_{i,2} = e(td_i, C_{i,2})$ and $X_i \| f(X_i) = C_{i,5} \oplus H_3(U_{i2})$. Thus, we can obtain Eq (3):

$$\begin{cases} f(X_1) = f_{1,0} + f_{1,1} X_1 + \cdots + f_{1,t-1} X_1^{t-1} \\ f(X_2) = f_{2,0} + f_{2,1} X_2 + \cdots + f_{2,t-1} X_2^{t-1} \\ \qquad \vdots \\ f(X_t) = f_{t,0} + f_{t,1} X_t + \cdots + f_{t,t-1} X_t^{t-1} \end{cases} \tag{3}$$

3) Let $f_{i,k} = f_{j,k}$, where $i, j \in \{1, 2, \cdots, t\}$ and $k \in \{0, 1, \cdots, t-1\}$. According to Cramer's rule and the properties of the Vandermonde matrix, we can obtain $\det(V) \neq 0$, thus obtaining the unique solution $f_{i,0}, f_{i,1}, \cdots, f_{i,t-1}$.

4) For each ciphertext δ_i, if $C_{i,6} = H_4(n_i \| C_{i,1} \| \cdots \| C_{i,5} \| U_{i,2} \| f_{i,0} \| \cdots \| f_{i,t-1})$ holds, meaning $m_1 = m_2 = \cdots = m_t$, then return with test result 1. Otherwise, return with result 0.

G) Unsigncryption

Given *params* and the identity ID_s of the DO, the DU downloads cyphertext δ from CS and executes the following operations:

1) Calculate $U_1^{'} = SK_{r2} C_1$, $U_2^{'} = e(SK_{r1}, C_2)$ and $m^{'} \| r_1^{'} = H_3(U_1^{'}) \oplus H_5(U_2^{'}) \oplus C_3$;

2) Calculate $t = H_1(m^{'})$ and test whether $g^{r_1^{'}} = e(C_4, H_1(ID_s)P + Pub_1)g^{t^{-1}}$ is true. If yes, execute the following calculation. Otherwise, return $\perp$;

3) Calculate $f_1^{'} = H_1(m^{'} \| n \| f_0^{'})$, $\cdots$, $f_{n-1}^{'} = H_1(m^{'} \| n \| f_0^{'} \| \cdots \| f_{n-2}^{'})$;

4) Calculate $X^{'} \| f^{'}(X^{'}) = C_5 \oplus H_3(U_2^{'})$. Test whether $f^{'}(X^{'}) = f_0^{'} + \cdots + f_{n-1}^{'} X^{'n-1}$ and $C_6 = H_4(n \| C_1 \| \cdots \| C_5 \| U_2^{'} \| f_0^{'} \| \cdots \| f_{n-1}^{'})$ are true. If both are true, return $m^{'}$. Otherwise, return $\perp$.

6. Smart contract

After the DU uploads the trapdoor to the blockchain, the smart contract completes the MET operation according to the pre-defined rules, and Algorithms 1 and 2.

```
Algorithm 1: Definition of data structure
Input: Define method
Output: Ciphertext_Tuple, DU_Tuple and Test_Tuple
Structure Ciphertext_Tuple {% Used to store ciphertext
Number int
$C_1, C_2, C_3, C_4, C_5, C_6$ string}
Structure DU_Tuple {% Used to store the identity and public key of DU
ID string
PK string}
Structure Test_Tuple {% Used to store the information required in the MET phase
ID string
Trappdoor string
Cipher Ciphertext_Tuple}
```

7. Correctness analysis

7.1. Correctness of unsigncryption

By calculating $m^{'} \| r_1^{'} = H_3(U_1^{'}) \oplus H_5(U_2^{'}) \oplus C_3$, the DU can get plaintext $m^{'}$, in which $U_1^{'} = SK_{r2}C_1$ and $U_2^{'} = e(SK_{r1}, C_2)$. Equations (4) and (5) hold:

$$U_1^{'} = SK_{r2}C_1 = SK_{r2}r_1P = r_1PK_r = U_1 \tag{4}$$

$$U_2^{'} = e(SK_{r1}, C_2) = e(s_2H_2(ID_r), r_2P) = e(Pub_2, Q_r)^{r_2} = U_2 \tag{5}$$

Then, we get $U_1^{'} = U_1$ and $U_2^{'} = U_2$, and Eq (6) holds:

$$\begin{aligned} m^{'} \| r_1^{'} &= H_3(U_1^{'}) \oplus H_5(U_2^{'}) \oplus C_3 \\ &= H_3(U_1^{'}) \oplus H_5(U_2^{'}) \oplus H_3(U_1) \oplus H_5(U_2) \oplus (m \| r_1) \\ &= m \| r_1 \end{aligned} \tag{6}$$

To sum up, the correctness of the unsigncryption is established.

```
Algorithm 2: MET
Input: Identities, trapdoors and ciphertexts
Output: Test result
Query (Index string)                 % It queries data by index value
Return Q[key] or  ⊥                  % Q is a map value of the form (key, value)
End Query
Update (flag, id, message, counter)
Retrieve the Test_Tuple x= Q [id || counter] of the user by (id, counter) from Q.
IF flag==1 THEN
    Set x.Trapdoor = message
ELSE
    Convert the message to the Ciphertext_Tuple data
    Set x.Cipher = data
    Set n= data.Number
END IF
END Update
Set M [id || counter] =  $x$
IF Each ciphertext corresponds to the same value of  $n_i$  THEN
    Test ($id_i \cdots id_j, td_i \cdots td_j$):
    While i < n
        do Retrieve the Test_Tuple  $x_i$ = Q [$id_i$] of the DU  $i$
        IF $x_i$ =⊥  THEN
            return null
        ELSE
            Parse  $x_i$ .Cipher into  $\delta_i$=($C_{i,1}, C_{i,2}, C_{i}{,}_{3}, C_{i,4}, C_{i}{,}_{5}, C_{i,6}, n_i$)
            Computes  $U_{i,2} = e(td_i, C_{i,2})$  and  $X_i \| f(X_i) = C_{i,5} \oplus H_3(U_{i2})$
        END IF
    END While
ELSE
    return null
END IF
Computes the unique solution  $f_{i,0}, f_{i,1}, \cdots, f_{i,n-1}$
IF  $C_{i,6} = H_4(n_i \| C_{i,1} \| \cdots \| C_{i,5} \| U_{i,2} \| f_{i,0} \| \cdots \| f_{i,t-1})$  is established for  $x_i$ .Cipher
    return 1
ELSE
    return 0
END IF
```

7.2. Correctness of signature verification

The DU can calculate $t = H_1(m')$, and further verify $g^{r'_1} = e(C_4, H_1(ID_s)P + Pub_1)g^{t^{-1}}$. Thus, we can obtain Eq (7):

$$\begin{aligned} g^{r_1'} &= e(C_4, H_1(ID_s)P + Pub_1)g^{t^{-1}} \\ &= e((r_1+t)SK_s, (H_1(ID_s)+s_1)P)g^{t^{-1}} \\ &= e((r_1+t)\frac{1}{H_1(ID_s)+s_1}P, (H_1(ID_s)+s_1)P)g^{t^{-1}} \\ &= g^{r_1} \end{aligned} \tag{7}$$

To sum up, the signature verification correctness of our scheme is established.

7.3. Correctness of MET

Given t ciphertexts δ_i=$(C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}, C_{i,5}, C_{i,6}, n_i)$, the blockchain verifies whether $n_1 = n_2 = \cdots = n_t = t$ is established. If yes, execute the following operation. If not, return $\perp$.

1) Calculate $U_{i2} = e(td_i, C_{i,2})$ and $X_i \,||\, f(X_i) = H_3(U_{i,2}) \oplus C_{i,5}$. Assume that the plaintexts are $m_1, m_2, \cdots, m_t$.

2) If $m_1 = m_2 = \cdots = m_t$, we can obtain $f_{i,k} = f_{j,k}$, where $i, j \in \{1, 2, \cdots, t\}$ and $k \in \{0, 1, \cdots, t-1\}$. Thus, we obtain Eq (8):

$$V = \begin{bmatrix} 1 & X_1 & X_1^2 & \cdots & X_1^{t-1} \\ 1 & X_2 & X_2^2 & \cdots & X_2^{t-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_t & X_t^2 & \cdots & X_t^{t-1} \end{bmatrix} \tag{8}$$

3) According to the randomness of X_i, the probability of $\det(V) \neq 0$ is $1 - \frac{1}{q(q-1)\cdots(q-t+1)}$. There is a unique solution $f_{i,0}{}^*, f_{i,1}{}^*, \cdots, f_{i,t-1}{}^*$. For t ciphertexts δ_i, it can establish $C_{i,6} = H_4(n_i \,||\, C_{i,1} \,||\cdots|\, C_{i,5} \,||\, U_{i,2} \,||\, f_{i,0}{}^* \,||\cdots||\, f_{i,t-1}{}^*)$.

4) If $m_1 \neq m_2 = \cdots = m_t$, we can obtain $f_{1,k} \neq f_{i,k} = f_{j,k}$, in which $i, j \in \{2, 3, \cdots, t\}$ and $k \in \{0, 1, \cdots, t-1\}$. Obviously, the solution can't establish Eqs (9) and (10) at the same time.

$$C_{1,6} = H_4(n_1 \,||\, C_{1,1} \,||\, C_{1,2} \,||\, C_{1,3} \,||\, C_{1,4} \,||\, C_{1,5} \,||\, U_{1,2} \,||\, f_{i,0}{}^* \,||\, f_{i,1}{}^* \,||\cdots||\, f_{i,t-1}{}^*) \tag{9}$$

$$C_{2,6} = H_4(n_2 \,||\, C_{2,1} \,||\, C_{2,2} \,||\, C_{2,3} \,||\, C_{2,4} \,||\, C_{2,5} \,||\, U_{2,2} \,||\, f_{i,0}{}^* \,||\, f_{i,1}{}^* \,||\cdots||\, f_{i,t-1}{}^*) \tag{10}$$

To sum up, the test algorithm is correct when the underlying plaintexts are either equal or not.

8. Security analysis

8.1. Reliability

When the MET transaction is released on the consortium blockchain platform, the smart contract will perform testing operations according to the predefined rules. Then, the test results are publicly recorded as contract state values in the consortium blockchain. As a result, everyone in the consortium chain can verify the test results. Based on this feature, the proposed scheme meets the credibility of the equivalent test results.

8.2. Confidentiality

In ROM, the proposed scheme satisfies the indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2) under type-I and type-II.

Theorem 1(IND-CCA2-1). In ROM, there is an adversary of type-I, A_1. After trapdoor queries of at most q_{td} times, q_{sk} times of private key queries, q_{sc} times of signcryption queries and q_{usc} times of unsigncryption queries, if A_1 can win the following game in polynomial time with a non-negligible advantage ε_1, then a challenger C can solve the BDHP with the probability $\varepsilon_1^{'}$ shown in Eq (11):

$$\varepsilon_1^{'} \geq \frac{1}{q}(1-\frac{q_{td}}{2^{\lambda}})(1-\frac{q_{sk}}{2^{\lambda}})(1-\frac{1}{q_{sc}+q_{sk}+1})^{q_{sc}+q_{sk}}(1-\frac{q_{usc}}{2^{\lambda}})\varepsilon_1 \tag{11}$$

Proof. C is the challenger to solve the BDHP problem. A_1 is a type-I adversary. Given (P,aP,bP,cP), where $a,b,c \in Z_q^*$, with A_1, C can calculate $e(P,P)^{abc}$.

Setup. C randomly selects $a \in Z_q^*$, calculates $Pub_2 = aP$, and returns *params* to A_1.

Phase 1. Initially, C maintains empty lists $L_i,(i=1,2,3,4,5)$ and records A_1 queries of $H_i,(i=1,2,3,4,5)$. A_1 can make the following queries.

1) H_1 **query:** A_1 submits ID_{ri} to inquire about the hashed value. C inquires whether there is an h_{1i} in L_1. If yes, return it to A_1. Otherwise, C randomly selects $h_{1i} \in Z_q^*$ and returns to A_1. Then, C inserts (ID_{ri},h_{1i}) into L_1.
2) H_2 **query:** A_1 submits ID_{ri} to inquire about the hashed value. C inquires whether there is an h_{2i} in L_2. If yes, return it to A_1. Otherwise, C randomly selects $h_{2i} \in G_1$ and returns to A_1. Then, C inserts (ID_{ri},h_{2i}) into L_2.
3) H_3 **query:** $\mathcal{A}_1$ submits $U_{i,1}$ to inquire about the hashed value. $\mathcal{C}$ inquires whether there is an h_{3i} in L_3. If yes, return it to $\mathcal{A}_1$. Otherwise, $\mathcal{C}$ randomly selects $h_{3i} \in \{0,1\}^{l_m+l_0}$ and returns to $\mathcal{A}_1$, then inserts $(U_{i,1},h_{3i})$ into L_3.
4) H_4 **query:** $\mathcal{A}_1$ submits $n_i \| C_{i,1} \| \cdots \| C_{i,5} \| U_{i,2} \| f_{i,0} \| \cdots \| f_{i,t-1}$ to inquire about the hashed value. $\mathcal{C}$ inquires whether there is h_{4i} in L_4. If yes, return it to $\mathcal{A}_1$. Otherwise, $\mathcal{C}$ randomly selects $h_{4i} = \{0,1\}^{\lambda}$, returns it to $\mathcal{A}_1$. $\mathcal{C}$ inserts $(n_i \| C_{i,1} \| \cdots \| C_{i,5} \| U_{i,2} \| f_{i,0} \| \cdots \| f_{i,t-1}, h_{4i})$ into L_4.
5) H_5 **query:** $\mathcal{A}_1$ submits $U_{i,2}$ to inquire about the hashed value. $\mathcal{C}$ inquires whether there is an h_{5i} in L_5. If yes, return it to $\mathcal{A}_1$. Otherwise, $\mathcal{C}$ randomly selects $h_{5i} \in \{0,1\}^{l_m+l_0}$ and returns it to $\mathcal{A}_1$. $\mathcal{C}$ inserts $(U_{i,2},h_{5i})$ into L_5.
6) **Partial private key query:** $\mathcal{A}_1$ makes the query to identity ID_{ri}. $\mathcal{C}$ executes the PPK-Gen algorithm to calculate D_{ri} and returns it to $\mathcal{A}_1$.
7) **Private key query:** $\mathcal{A}_1$ makes the query to identity ID_{ri}. $\mathcal{C}$ executes the SV-set and Key-Gen

algorithms to calculate SK_{ri}. Then, C returns it to $\mathcal{A}_1$.

8) **Public key query:** $\mathcal{A}_1$ makes the query to identity ID_{ri}. C executes the Key-Gen algorithm to calculate PK_{ri} and returns it to $\mathcal{A}_1$.
9) **Replace public key query:** $\mathcal{A}_1$ can select PK_{ri}^* to replace the pubic key PK_{ri} of ID_{ri}.
10) **Signcryption query:** $\mathcal{A}_1$ selects the sender's identity ID_{si}, the plaintext m_i and the receiver identity ID_{ri}. C executes the signcryption algorithm to calculate ciphertext δ_i and returns it to $\mathcal{A}_1$. Besides, if the public key PK_{ri} has been replaced by $\mathcal{A}_1$, $\mathcal{A}_1$ needs to provide C with the secret value.
11) **Unsigncryption query:** $\mathcal{A}_1$ selects the sender's identity ID_{si}, the receiver's identity ID_{ri} and the ciphertext δ_i. C executes the unsigncryption algorithm to calculate the plaintext m_i and returns it to $\mathcal{A}_1$.

Challenge. $\mathcal{A}_1$ can decide when to stop phase 1 and enter this. $\mathcal{A}_1$ selects the sender's identity ID_s^*, the receiver's identity ID_r^* and two plaintext messages of equal length, m_0 and m_1. $\mathcal{A}_1$ has not made a private key query to ID_r^*. C randomly selects $\xi \in \{0,1\}$ and executes the following calculation.

1) Calculate $f_0^* = H_1(m_\xi \| n)$, $f_1^* = H_1(m_\xi \| n \| f_0^*)$, $\cdots$, $f_{n-1}^* = H_1(m_\xi \| n \| f_0^* \| \cdots \| f_{n-2}^*)$ and $f^*(x) = f_0^* + f_1^* x + \cdots + f_{n-1}^* x^{n-1}$.
2) Randomly select $r_1, b, c, X \in \mathrm{Z}_q^*$.
3) Calculate $U_1^* = r_1 PK_r^*$, $Q_r^* = bP$ and $U_2^* = e(Pub_2, Q_r^*)^c$.
4) Calculate $C_1^* = r_1 P$, $C_2^* = cP$, and $C_3^* = (m_\xi \| r_1) \oplus H_3(U_1^*) \oplus H_5(U_2^*)$.
5) Calculate $C_4^* = (H_1(m_\xi) + r_1) SK_s^*$, $C_5^* = (X \| f^*(X)) \oplus H_3(U_2^*)$ and $C_6^* = H_4(n \| C_1^* \| \cdots \| C_5^* \| U_2^* \| f_0^* \| \cdots \| f_{n-1}^*)$.
6) Return $\delta^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*, n)$ to $\mathcal{A}_1$.

Phase 2. After receiving δ^*, $\mathcal{A}_1$ continues to make queries in phase 1. However, $\mathcal{A}_1$ can neither query the private key of ID_r^* nor make the unsigncryption query about $(\delta^*, ID_s^*, ID_r^*)$.

Guess. $\mathcal{A}_1$ outputs the guessed value $\xi^* \in \{0,1\}$. If $\xi^* = \xi$, then $\mathcal{A}_1$ wins the game. C will select $(U_2^*, H_5(U_2^*))$ in L_5 and take $U_2^* = e(Pub_2, Q_r^*)^c = e(aP, bP)^c = e(P,P)^{abc}$ as the solution to BDHP. However, the problem is intractable in the polynomial time. Then, the proposed scheme meets the IND-CCA2 security requirements under type-I attacks.

Theorem 2 (IND-CCA-2). In ROM, there is an adversary of type-II, $\mathcal{A}_2$. After the trapdoor queries of at most q_{td} times, q_{sk} times of private key queries, q_{sc} times of signcryption queries and q_{usc} times of unsigncryption queries, if $\mathcal{A}_2$ can win the following game process in polynomial time with a non-negligible advantage ε_2, then a challenger C can solve the CDHP problem with the probability $\varepsilon_2^{'}$ shown in Eq (12):

$$\varepsilon_2^{'} \geq \frac{1}{q}(1-\frac{q_{td}}{2^\lambda})(1-\frac{q_{sk}}{2^\lambda})(1-\frac{1}{q_{sc}+q_{sk}+1})^{q_{sc}+q_{sk}}(1-\frac{q_{usc}}{2^\lambda})\varepsilon_2 \tag{12}$$

Proof. C is the challenger to solve the CDHP problem. $\mathcal{A}_2$ is a type-II adversary. Given

(P, aP, bP), in which $a, b \in \mathrm{Z}_q^*$, with $\mathcal{A}_2$, then C can get abP.

Setup. C executes the setup algorithm and returns the master key s_2 and system parameter $params$ to $\mathcal{A}_2$.

Phase 1. $\mathcal{A}_2$ can execute other queries except replace the public key query in Theorem 1.

Challenge. $\mathcal{A}_2$ can decide when to stop phase 1 and enter this. $\mathcal{A}_2$ selects the sender's identity ID_s^*, the receiver's identity ID_r^*, and two plaintexts of equal length, m_0 and m_1. $\mathcal{A}_2$ has not made the private key queries and replace public key queries about ID_r^*. When $\mathcal{A}_2$ makes a public key query about ID_r^*, then C randomly selects $a \in \mathrm{Z}_q^*$, calculates $PK_r^* = aP$ and returns to $\mathcal{A}_2$. C randomly selects $\xi \in \{0,1\}$ and executes the following calculation.

1) Calculate $f_0^* = H_1(m_\xi \| n)$, $f_1^* = H_1(m_\xi \| n \| f_0^*)$, $\cdots$, $f_{n-1}^* = H_1(m_\xi \| n \| f_0^* \| \cdots \| f_{n-2}^*)$ and let $f^*(x) = f_0^* + f_1^* x + \cdots + f_{n-1}^* x^{n-1}$.
2) Randomly select $b, r_2, X \in \mathrm{Z}_q^*$.
3) Calculate $U_1^* = bPK_r^*$, $Q_r^* = H_2(ID_r^*)$ and $U_2^* = e(Pub_2, Q_r^*)^{r_2}$.
4) Calculate $C_1^* = bP$, $C_2^* = r_2 P$, $C_3^* = (m_\xi \| r_1) \oplus H_3(U_1^*) \oplus H_5(U_2^*)$, $C_4^* = (H_1(m_\xi) + r_1) SK_s^*$, $C_5^* = (X \| f^*(X)) \oplus H_3(U_2^*)$ and $C_6^* = H_4(n \| C_1^* \| \cdots \| C_5^* \| U_2^* \| f_0^* \| \cdots \| f_{n-1}^*)$.
5) Return $\delta^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*, n)$ to $\mathcal{A}_2$.

Phase 2. After receiving δ^*, $\mathcal{A}_2$ continues with the queries of phase 1; howwver, $\mathcal{A}_2$ can neither query about the private key of ID_r^* nor make an unsigncryption query about $(\delta^*, ID_s^*, ID_r^*)$.

Guess. $\mathcal{A}_2$ outputs the guessed value $\xi^* \in \{0,1\}$. If $\xi^* = \xi$, $\mathcal{A}_2$ wins the game. C will select $(U_1^*, H_3(U_1^*))$ from L_3 and take $U_1^* = bPK_r^* = abP$ as the solution to the CDHP problem. However, the problem is intractable in the polynomial time; therefore, the proposed scheme satisfies IND-CCA2 security under type-II attacks.

8.3. One way security

In ROM, the scheme in this paper satisfies the one way against an adaptive chosen ciphertext attacks (OW-CCA2) under type-III and type-IV.

Theorem 3 (OW-CCA2-1). In ROM, there is an adversary of type-III, $\mathcal{A}_3$. After q_{sk} times of private key queries, q_{sc} times of signcryption queries and q_{usc} times of unsigncryption queries, if $\mathcal{A}_3$ can win the relevant game in polynomial time with a non-negligible advantage ε_3, then a challenger C can solve the BDHP with the probability $\varepsilon_3^{'}$ shown in Eq (13):

$$\varepsilon_3^{'} \geq \frac{1}{q}\left(1 - \frac{q_{sk}}{2^\lambda}\right)\left(1 - \frac{1}{q_{sc} + q_{sk} + 1}\right)^{q_{sc} + q_{sk}}\left(1 - \frac{q_{usc}}{2^\lambda}\right)\varepsilon_3 \tag{13}$$

The proof process is similar to Theorem 1. We will not repeat it here.

Theorem 4 (OW-CCA2-2). In ROM, there is an adversary of type-IV, $\mathcal{A}_4$. After q_{sk} times of private key queries, q_{sc} times of signcryption queries and q_{usc} times of unsigncryption queries, if $\mathcal{A}_4$ can win the relevant game in polynomial time with a non-negligible advantage ε_4, then a challenger C can solve the CDHP with the probability $\varepsilon_4^{'}$ shown in Eq (14):

$$\varepsilon_4^{'} \geq \frac{1}{q}(1-\frac{q_{sk}}{2^{\lambda}})(1-\frac{1}{q_{sc}+q_{sk}+1})^{q_{sc}+q_{sk}}(1-\frac{q_{usc}}{2^{\lambda}})\varepsilon_4 \tag{14}$$

The proof process is similar to Theorem 2, so we will not repeat it here.

8.4. Unforgeability

In ROM, the proposed scheme meets the requirements of the existential unforgerability against chosen message attack (EUF-CMA).

Theorem 5 (EUF-CMA). In ROM, if there is an adversary $\mathcal{F}$ after q_{sk} times of key queries, q_{sc} times of signcryption queries and q_{usc} times of unsigncryption queries, with non-negligible advantage ε_5, it wins the following game process in polynomial time. Then, a challenger C can solve the q-SDHP problem with the probability $\varepsilon_5^{'}$ shown in Eq (15):

$$\varepsilon_5^{'} \geq (1-\frac{q_{sk}}{2^{\lambda}})(1-\frac{1}{q_{sc}+q_{sk}+1})^{q_{sc}+q_{sk}}\varepsilon_5 \tag{15}$$

Proof. C is a challenger to solve the q-SDHP problem. $\mathcal{F}$ is an adversary. Given $(P, aP, a^2P, \cdots, a^qP)$, in which $a, q \in Z_q^*$, with $\mathcal{F}$, then C can calculate $(k, \frac{1}{k+a}P)$.

Setup. C randomly selects $k_1, k_2, \cdots, k_{q-1}, a \in Z_q^*$. C calculates $f(x)=\prod_{i=1}^{q-1}(x+k_i)=\sum_{i=1}^{q-1}z_ix^i$, $P^{'}=\sum_{i=1}^{q-1}z_i(a^iP)=f(a)P \in G_1$ and $Pub_1=\sum_{i=1}^{q}z_{i-1}(a^iP)=aP^{'}$. Then, C returns identity ID_s^* and $params$ to $\mathcal{F}$, and secretly saves a.

Training. C maintains initially empty lists L_{sk} and $L_i (i=1,2,3,4,5)$ to record the results of the key and $H_i, (i=1,2,3,4,5)$ queries made by $\mathcal{F}$. The same query process as in theorem 1 will not be repeated here.

1) H_1 **query:** $\mathcal{F}$ submits ID_{si} to inquire the hashed value. If $ID_{si}=ID_s^*$, C randomly selects $k^* \in Z_q^*$ and $k^* \notin \{k_1, k_2, \cdots, k_{q-1}\}$. C returns it to $\mathcal{F}$, and inserts (ID_s^*, k^*) into L_1. If $ID_{si} \neq ID_s^*$, then C randomly selects $k_i \in \{k_1, k_2, \cdots, k_{q-1}\}$ and returns to $\mathcal{F}$. Finally, C inserts (ID_{si}, k_i) into L_1.

2) **Key query:** $\mathcal{F}$ makes the query about the identity ID_{si}. If $ID_{si}=ID_s^*$, then C terminates the query and returns with $\perp$. If not, C inquires L_1 to obtain (ID_{si}, k_i). C calculates $SK_i=\frac{1}{k_i+a}P^{'}$ and returns to $\mathcal{F}$, then inserts (ID_{si}, k_i, SK_i) into L_{sk}.

3) **Signcryption query:** $\mathcal{F}$ makes this query about (ID_{si}, m_i, ID_{ri}). If $ID_{si}=ID_s^*$, then C terminates the query and returns with '$\perp$'. If not, C inquires L_{sk} to obtain (ID_{si}, k_i, SK_i), and executes the following steps.

1) Calculate $f_{i,0}=H_1(m_i \| n)$, $\cdots$, $f_{i,n-1}=H_1(m_i \| n \| f_{i,0} \| \cdots \| f_{i,n-2})$, and let

$f_i(x) = f_{i,0} + f_{i,1}x + \cdots + f_{i,n-1}x^{n-1}$.

2) Randomly select $r_{i,1}, r_{i,2}, X_i \in Z_q^*$.
3) Calculate $U_{i,1} = r_{i,1}PK_{ri}$, $Q_{ri} = H_2(ID_{ri})$ and $U_{i,2} = e(Pub_2, Q_{ri})^{r_{i,2}}$.
4) Calculate $t_i = H_1(m_i)$, $C_{i,1} = r_{i,1}P$, $C_{i,2} = r_{i,2}P$, $C_{i,3} = (m_i \| r_{i,1}) \oplus H_3(U_{i,1}) \oplus H_5(U_{i,2})$, $C_{i,4} = (r_{i,1} + t_i)SK_i$, $C_{i,5} = (X_i \| f(X_i)) \oplus H_3(U_{i,2})$ and $C_{i,6} = H_4(n \| C_{i,1} \| \cdots \| C_{i,5} \| U_{i,2} \| f_{i,0} \| \cdots \| f_{i,n-1})$.
5) Return δ_i=$(C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}, C_{i,5}, C_{i,6}, n)$ to $\mathcal{F}$.

Forgery. According to the forking lemma [26], F can forge signatures (ID_s^*, m_i, t, C_4) and $(ID_s^*, m_i, t^*, C_4^*)$ in which $t \neq t^*$.

1) $\mathcal{C}$ calculate $C_4^* - C_4 = (r_1^* + t^*)SK_s^* - (r_1 + t)SK_s^*$, get $\frac{1}{a+k_s}P^{'} = h(C_4^* - C_4)$, in which $h = r_1^* + t^* - r_1 - t$.
2) $\mathcal{C}$ calculate $\frac{f(a)}{a+k_s}P = h^{-1}(C_4^* - C_4)$.
3) $\mathcal{C}$ calculate $f(x) = g(x)(x+k_s) + g_{-1}$, in which $g(x) = \sum_{i=0}^{q-2} z_i x^i$, $g_{-1} \in Z_q^*$.

Through the above calculations, we can obtain Eqs (16) and (17):

$$\frac{f(x)}{x+k_s} = \sum_{i=0}^{q-2} z_i x^i + \frac{g_{-1}}{x+k_s} \tag{16}$$

$$\frac{1}{a+k_s}P = (\frac{C_4^* - C_4}{h} - \sum_{i=0}^{q-2} z_i(a^i P)) + g_{-1}{}^{-1} \tag{17}$$

That is, $\mathcal{C}$ can output $(k_s, \frac{1}{a+k_s}P)$ as a solution of the q-SDHP problem. However, the problem is intractable in the polynomial time. The proposed scheme meets the EUF-CMA security.

9. Discussion

9.1. Function comparison

We compared our scheme with the schemes in references [26,27,29] in terms of function and security, and the results are shown in Table 1.

Compared with scheme in [29], by introducing HSC, our scheme realizes the heterogeneous communication from IBC to CLC. Our scheme achieves the higher securities of IND-CCA2, OW-CCA2, and EUF-CMF. The schemes in references [26,27,29] introduced ET technology; however, their test method is to group ciphertexts in pairs and then test one by one, thereby resulting in a high computational overhead. In our scheme, MET technology is introduced, which enables a tester to test *n* ciphertexts at the same time, reduces the computational cost of testers, improves the retrieval efficiency, and is thus more suitable for multi-user scenarios. In addition, compared with the scheme in reference [26], the scheme in this paper adopts a CLC and avoids certificate management problems in this way. Compared with the scheme in reference [27], the proposed scheme achieves a higher IND-

CCA2 and OW-CCA2 security.

Table 1. Comparison of function.

Scheme	ET	MET	HSC	Credibility	Confidentiality	One way security	Unforgeability
[26]	√	×	√	×	IND-CCA2	OW-CCA2	EUF-CMF
[27]	√	×	√	×	IND-CCA2	OW-CCA	EUF-CMF
[29]	√	×	×	×	IND-CCA	OW-CCA	×
Ours	√	√	√	√	IND-CCA2	OW-CCA2	EUF-CMF

9.2. Comparison of computational overhead

To evaluate the computing performance, we use the operating system equipped with i7-7500u, 3.5GHz processor, 8G memory, and Windows 10 for a simulation with PBC library in VC6.0 environment. The average value of 50 simulations is taken as the result. The representative symbols and descriptions are shown in Table 2.

Table 2. Symbol and description.

Symbol	Meaning	Time(ms)
n	Number of plaintexts/ciphertexts	–
T_h	Time of an ordinary hash operation	0.0001
T_m	Time of a Map-to-Point hash operation	0.1521
T_a	Time of a scalar multiplication operation	1.3667
T_e	Time of an exponential operation	4.2511
T_p	Time of a bilinear pairing operation	9.4326

We compared the computational overhead of our scheme with the schemes in [26,27] at the signcryption, unsigncryption and test phases. The results are shown in Table 3. At each phase, with the increase of the number of plaintexts/ciphertexts, the computational overhead is shown in Figures 2, 3, and 4, respectively.

Table 3. Comparison of computational cost.

Scheme	Signcryption	Unsigncryption	Test
[26]	$3nT_h + 2nT_a + nT_e + 3nT_p$ $= 35.2826n$	$4nT_h + 2nT_a + nT_e + 3nT_p$ $= 35.2827n$	$n(n-1)(T_h + T_a + T_e + T_p)$ $= 15.0505(n^2 - n)$
[27]	$3nT_h + nT_m + 5nT_a + 2nT_e + 2nT_p$ $= 34.3533n$	$3nT_h + nT_m + 5nT_a + 2nT_p$ $= 25.8511n$	$n(n-1)(T_h + T_a + 2T_p)$ $= 20.2320(n^2 - n)$
Ours	$n(n+4)T_h + nT_m + 4nT_a + nT_e + nT_p$ $= 0.0001n^2 + 19.3030n$	$n(n+5)T_h + 2nT_a + 3nT_e + 2nT_p$ $= 0.0001n^2 + 34.3524n$	$2nT_h + nT_p = 9.4328n$

It can be seen from Table 3 and Figures 2, 3 and 4 that compared with the schemes in [26,27], our scheme reduces the bilinear pairing operation with a large computational overhead in signcryption phase. Although it involves more hash computation, the time cost of the hash is very small. Therefore, the computational overhead of our scheme in the signcryption phase is lower than schemes in [26,27]. In the unsigncryption phase, the computational overhead of our scheme is lower than that of the scheme in [26]. Compared with the scheme in reference [27], the computational overhead is large; however, in the previous subsection, we have proved that our scheme has a higher security than that in reference [27]. In the ET phase, the schemes in references [26,27] only support grouping ciphertexts in pairs and then performing the ET. With the increase of ciphertexts, the computational cost dramatically increases. In our scheme, a tester can test multiple ciphertexts equivalently at the same time, which greatly reduces the overhead.

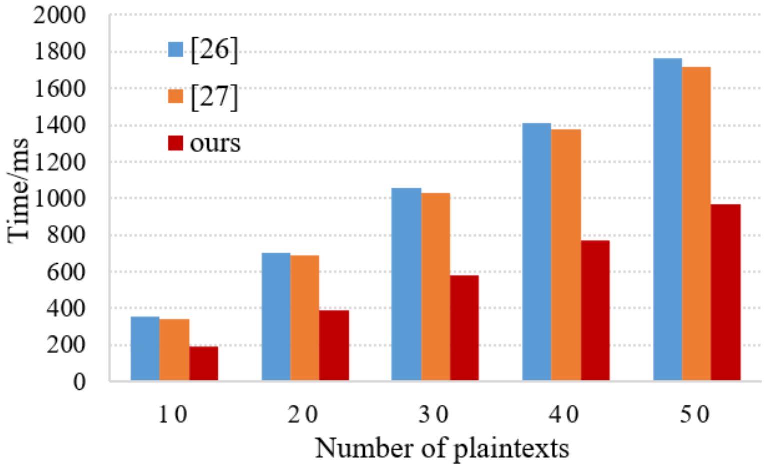

Figure 2. Computational overhead at signcryption phase.

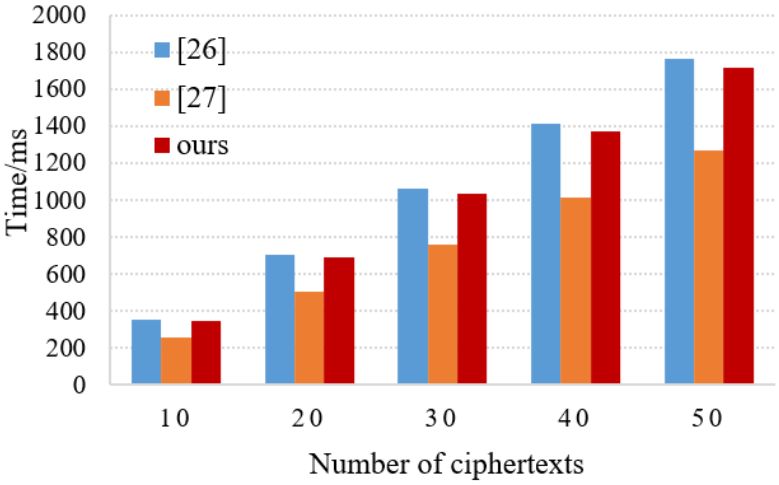

Figure 3. Computational overhead at unsigncryption phase.

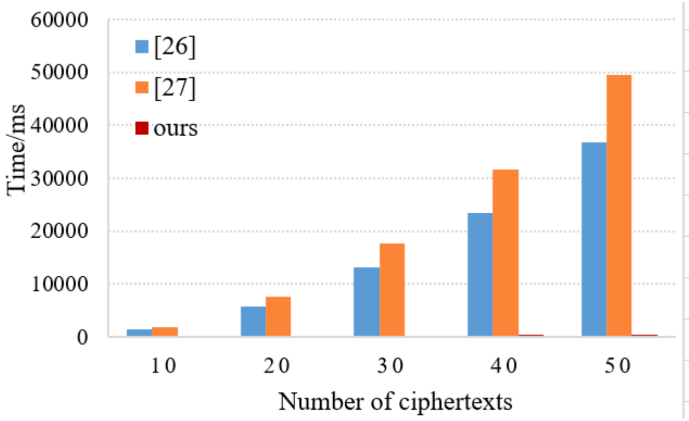

Figure 4. Computational overhead at test phase.

9.3. The limitations of the solution

In the real world, the adoption of this solution may face the following challenges and obstacles.

Technical compatibility: Smart grid solutions typically involve multiple technologies. Ensuring the compatibility between different technologies and integration with existing devices may pose certain challenges, thus requiring specific planning and testing.

Investment cost: Implementing this solution in a smart grid requires significant investment, thus necessitating thorough economic evaluation and planning.

Relevant policies and regulations: Smart grids involve aspects such as data privacy and market transactions, among others. It is necessary to align with relevant policies and regulations. System design, operation, and data usage need to be compliant to ensure adherence to legal requirements.

10. Conclusions

This paper proposed an HSC scheme for the smart grid with a trusted MET. The adoption of an HSC realizes the secure communication from IBC to CLC, avoids the limitation of a single cryptosystem scheme, and ensures the confidentiality, integrity and authentication of user power data in the smart grid. The introduction of MET technology enables testers to retrieve multiple ciphertexts safely and efficiently at the same time, thereby greatly reducing the computational overhead. The use of blockchains and smart contracts ensures the credibility of test results and eliminates the dependence of equivalent test operations on trusted CSs. We compared the proposed scheme with some similar schemes. The results show that the scheme in this paper has higher security attributes, additional functional features, and a lower computational overhead. However, its security is proven under the ROM. In future works, we will aim to design a scheme in the standard model, and further optimize the algorithm, protocol and architecture design to improve the scalability of the system.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (nos. 62362059, 62172337), the Industrial Support Plan Project of Gansu Provincial Education Department (no. 2023CYZC-09) and the Key Research and Development Program of Gansu Province (no. 23YFGA0081).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. J. Zhang, P. Dai, T. Wu, H. Yu, B. Lin, B. Qi, Architecture design and demand analysis of new generation technical standard system for smart grid, *Autom. Electr. Power Syst.*, **44** (2020), 12–20.
2. O. M. Butt, M. Zulqarnain, T. M. Butt, Recent advancement in smart grid technology: Future prospects in the electrical power network, *Ain Shams Eng. J.,* **12** (2021), 687–695. https://doi.org/10.1016/j.asej.2020.05.004
3. S. K. Rathor, D. Saxena, Energy management system for smart grid: an overview and key issues, *Int. J. Energy Res.,* **44** (2020), 4067–4109. https://doi.org/10.1002/er.4883
4. C. Ren, P. Li, Y. Bai, B. Xu, A resource allocation method of smart grid based on heterogeneous network, *Inf. Technol.*, **12** (2021), 89–94. https://doi.org/10.13274/j.cnki.hdzj.2021.12.016
5. T. Dillon, C. Wu, E. Chang, Cloud computing: issues and challenges, in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, (2010), 27–33. https://doi.org/10.1109/AINA.2010.187
6. D. Boneh, G. D. Crescenzo, R. Ostrovsky, G. Persiano, Public key encryption with keyword search, in *International Conference on the Theory and Applications of Cryptographic Techniques*, (2004), 506–522. https://doi.org/10.1007/978-3-540-24676-3_30
7. G. Yang, C. H. Tan, Q. Huang, D. S. Wong, Probabilistic public key encryption with equality test, in *Cryptographers' Track at the RSA Conference*, (2010), 119–131. https://doi.org/10.1007/978-3-642-11925-5_9
8. H. T. Lee, S. Ling, J. H. Seo, H. Wang, T. Youn, Public key encryption with equality test in the standard model, *Inf. Sci.*, **516** (2020), 89–108. https://doi.org/10.1016/j.ins.2019.12.023
9. G. G. Deverajan, V. Muthukumaran, C. Hsu, M. Karuppiah, Y. Chung, Y. Chen, Public key encryption with equality test for industrial internet of things system in cloud computing, *Trans. Emerging Telecommun. Technol.*, **33** (2022), 4202. https://doi.org/10.1002/ett.4202
10. H. Zhu, L. Wang, H. Ahmad, D. Xie, Pairing-free for public key encryption with equality test scheme, *IEEE Access*, **9** (2021), 77239–77249. https://doi.org/10.1109/ACCESS.2021.3081709
11. W. Susilo, F. Guo, Z. Zhao, G. Wu, PKE-MET: Public-key encryption with multi-ciphertext equality test in cloud computing, *IEEE Trans. Cloud Comput.*, **10** (2022), 1476–1488. https://doi.org/10.1109/TCC.2020.2990201
12. Y. Zheng, Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost(signature) + cost (encryption), in *Annual International Cryptology Conference*, **1924** (1997), 165–179. https://doi.org/10.1007/BFb0052234

13. H. Xiong, Y. Hou, X. Huang, Y. Zhao, Secure message classification services through identity-based signcryption with equality test towards the internet of vehicles, *Veh. Commun.*, **26** (2020), 100264. https://doi.org/10.1016/j.vehcom.2020.100264
14. Y. Zhang, Q. Bai, Y. Ma, C. Yan, C. Wang, Certificateless signcryption with equality test, *J. Electron. Inf. Technol.*, **43** (2021), 2534–2541.
15. X. Yang, H. Zhou, Z. Wang, S. Yuan, C. Wang, Blockchain-based certificateless signcryption scheme with equality test for wireless body area network, *Acta Electron. Sin.*, **51** (2023), 922–932.
16. H. Xiong, Y. Zhao, Y. Hou, X. Huang, C. Jin, L. Wang, et al., Heterogeneous signcryption with equality test for IoT environment, *IEEE Int. Things J.*, **8** (2021), 16142–16152. https://doi.org/10.1109/JIOT.2020.3008955
17. X. G. Shan, J. Zhuang, A game-theoretic approach to modeling attacks and defenses of smart grids at three levels, *Reliab. Eng. Syst. Saf.*, **195** (2020), 106683. https://doi.org/10.1016/j.ress.2019.106683
18. Q. Tang, Towards public key encryption scheme supporting equality test with fine-grained authorization, in *Australasian Conference on Information Security and Privacy*, **6812** (2011), 389–406. https://doi.org/10.1007/978-3-642-22497-3_25
19. Q. Tang, Public key encryption schemes supporting equality test with authorisation of different granularity, *Int. J. Appl. Cryptogr.*, **2** (2012), 304–321. https://doi.org/10.1504/IJACT.2012.048079
20. Q. Tang, Public key encryption supporting plaintext equality test and user-specified authorization, *Secur. Commun. Networks*, **5** (2012), 1351–1362. https://doi.org/10.1002/sec.418
21. N. Li, Efficient equality test on identity-based ciphertexts supporting flexible authorization, *Entropy*, **25** (2023), 362. https://doi.org/10.3390/e25020362
22. P. S. Roy, D. H. Duong, W. Susilo, A. Sipasseuth, K. Fukushima, S. Kiyomoto, Lattice-based public-key encryption with equality test supporting flexible authorization in standard model, *Theor. Comput. Sci.*, **929** (2022), 124–139. https://doi.org/10.1016/j.tcs.2022.06.034
23. X. Lin, L. Sun, H. Qu, X. Zhang, Public key encryption supporting equality test and flexible authorization without bilinear pairings, *Comput. Commun.*, **170** (2021), 190–199. https://doi.org/10.1016/j.comcom.2021.02.006
24. S. Ma, Identity-based encryption with outsourced equality test in cloud computing, *Inf. Sci.*, **328** (2016), 389–402. https://doi.org/10.1016/j.ins.2015.08.053
25. H. Qu, Z. Yan, X. Lin, Q. Zhang, L. Sun, Certificateless public key encryption with equality test, *Inf. Sci.*, **462** (2018), 76–92. https://doi.org/10.1016/j.ins.2018.06.025
26. H. Xiong, Y. Hou, X. Huang, Y. Zhao, C. Chen, Heterogeneous signcryption scheme from IBC to PKI with equality test for WBANs, *IEEE Syst. J.*, **16** (2022), 2391–2400. https://doi.org/10.1109/JSYST.2020.3048972
27. Y. Hou, X. Huang, Y. Chen, S. Kumar, H. Xiong, Heterogeneous signcryption scheme supporting equality test from PKI to CLC toward IoT, *Trans. Emerging Telecommun. Technol.*, **32** (2021), 4190. https://doi.org/10.1002/ett.4190
28. Y. Zhao, Y. Hou, Y. Chen, S. Kumar, F. Deng, An efficient certificateless public key encryption with equality test toward internet of vehicles, *Trans. Emerging Telecommun. Technol.*, **33** (2022), 3812. https://doi.org/10.1002/ett.3812

29. X. Yang, H. Zhou, N. Ren, S. Yuan, C. Wang, An aggregated signcryption scheme for wireless body area networks supporting multi-ciphertext equivalence tests, *J. Comput. Res. Dev.*, **2** (2023), 341–350. https://doi.org/10.7544/issn1000-1239.202110775

30. J. Kim, K. H. Lee, J. Kim, Linking blockchain technology and digital advertising: How blockchain technology can enhance digital advertising to be more effective, efficient, and trustworthy, *J. Bus. Res.*, **160** (2023), 113819. https://doi.org/10.1016/j.jbusres.2023.113819

AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)