HSC-MET: Heterogeneous signcryption scheme supporting multi-ciphertext equality test for Internet of Drones

Xiaodong Yang, Ningning Ren, Aijia Chen, Zhisong Wang, Caifen Wang

1 Department of Computer Science and Engineering, Northwest Normal University, Lanzhou, Gansu, China, 2 Department of Big Data and Internet, Shenzhen Technology University, Shenzhen, Guangdong, China

Abstract

Internet of Drones (IoD) is considered as a network and management architecture, which can enable unmanned aerial vehicles (UAVs) to collect data in controlled areas and conduct access control for UAVs. However, the current cloud-assisted IoD scheme cannot efficiently achieve secure communication between heterogeneous cryptosystems, and does not support multi-ciphertext equality tests. To improve the security and performance of traditional schemes, we propose a heterogeneous signcryption scheme (HSC-MET) that supports multi-ciphertext equality test. In this paper, we use a multi-ciphertext equality test technique to achieve multi-user simultaneous retrieval of multiple ciphertexts safely and efficiently. In addition, we adopt heterogeneous signcryption technology to realize secure data communication from public key infrastructure (PKI) to certificateless cryptography (CLC). At the same time, the proposed scheme based on the computation without bilinear pairing, which greatly reduces the computational cost. According to the security and performance analysis, under the random oracle model (ROM), the confidentiality, unforgeability and number security of HSC-MET are proved based on the computational Diffie-Hellman (CDH) problem.

Introduction

Unmanned aerial vehicles (UAVs) [1, 2] as devices using radio remote control technology and self-provided program control mechanism, have the advantages of small size, low cost, and flexible deployment. As a result, it is widely used in film and television shooting, environmental monitoring, and smart farms. To provide coordinated and orderly access for UAVs, the Internet of Drones (IoD) [3–5] came into being. IoD is a sophisticated heterogeneous network containing a large number of sensors and actuators. In IoD environment, entities communicate through open wireless channels, thus facing many privacy and security issues [6]. Entities in IoD also have limited computing and storage capabilities, so it is extremely important to design an efficient and secure algorithm. Bharany et al. [7] proposed a clustering protocol for flying ad-hoc networks (FANETs) based on a moth flame optimization algorithm for safe and
efficient UAV access. It ensures UAVs’ efficient and safe access while also improving FANET fault tolerance. Bharany et al. [8] proposed a unique clustering algorithm EE-SS for FANETs to increase the service life of UAVs in forest fire detection, which reduced cluster head overhead and improved system efficiency. With the wide application of UAVs, the storage and processing of big data in IoD have become a top priority. Fortunately, cloud computing technology can provide users with computing services regardless of time and place. However, since cloud servers are not trusted, data is usually encrypted or signcrypted and stored in cloud servers, which makes efficient data retrieval difficult.

To ensure the security of UAVs, Bera et al. [9] proposed a blockchain-based secure access control scheme to achieve authentication between drones and between drones and a ground station server. The scheme satisfies the immutability of data. Hussain et al. [10] proposed an authentication scheme based on elliptic curve cryptography to secure the communication between a data user and a drone. Khan et al. [11] proposed an identity-based proxy signcryption scheme based on hyperelliptic curves. The scheme allows for outsourced decryption to reduce the computational cost. They proved that the scheme satisfies indistinguishability against adaptive selected scrambled text attacks and existential forgery for adaptive selected plaintext attacks under the ROM. Gope and Sikdar [12] proposed an efficient privacy-aware authenticated key agreement scheme for edge-assisted IoD. The scheme does not need to store any secret keys in the devices but still can provide the desired security features. But the IoD is a heterogeneous and complex network, so these schemes in [9–12] are inapplicable. To realize secure communication between heterogeneous cryptosystems, Sun and Li [13] proposed a heterogeneous signcryption scheme (HSC), which realized the secure communication from public key infrastructure (PKI) to identity-based cryptography (IBC). Inspired by Sun and Li, many HSC schemes have been proposed [14–20].

Although the schemes proposed in [14–20] have realized the secure communication between heterogeneous cryptosystems, it does not consider the efficient retrieval of ciphertexts. Cloud storage has brought great convenience, but this approach reduces the availability of data. Boneh et al. [21] proposed to use keyword search-based public key encryption (PKE-KS) to realize ciphertext retrieval in cloud servers, but it only supports retrieval of ciphertext encrypted with the same public key. To improve this limitation, Yang et al. [22] proposed a public key encryption scheme that supports the ciphertext equality test (PKE-ET), which allows users to compare two ciphertexts obtained by using the different public keys. Subsequently, scholars have proposed a series of similar schemes [23–27], but these schemes only support the equality test after dividing two ciphertexts into a group. Therefore, it faces the challenges of low retrieval efficiency and high computational cost. To reduce computational cost and improve the efficiency of ciphertext retrieval, Susilo et al. [28] proposed public-key encryption with flexible multi-ciphertext equality test (PKE-FMET). Although this scheme supports the equality test of more than two ciphertexts, there are problems such as not satisfying message authentication and communication between heterogeneous cryptosystems.

**Our contributions**

With the motivation of solving the above-mentioned problems, we present a heterogeneous signcryption scheme that supports the multi-ciphertext equality test (HSC-MET). The main contributions are as follows.

1. Our scheme utilizes heterogeneous signcryption technology to realize secure communication from PKI to certificateless public key cryptography (CLC), eliminating the limitation of existing schemes that only support communication in the same cryptosystem.
2. We adopt the multi-ciphertext equality testing technique to address the limitations of pairwise ciphertext equality testing to reduce the computational cost required for ciphertext equality testing in multi-user and multi-ciphertext environments.

3. Our scheme is based on computation without bilinear pairing, which greatly reduces the computing cost and improves the communication and retrieval efficiency for the problem of limited computing resources of UAVs.

4. Our scheme is proven to meet unforgeability and confidentiality based on the CDH problem under the ROM. We demonstrated our scheme’s number security using the definition of a new security number-security proposed in [26].

5. We compared our scheme with similar schemes in terms of confidentiality, unforgeability, and computational costs. Analysis results show that our scheme meets higher confidentiality, unforgeability and lower computational costs.

Organization
The rest of this paper is structured as follows. The complexity assumption, Kramer’s rule, Vandermonde determinant, formal definition, and security design are all introduced in section 2. The system design is presented in section 3. In section 4, we go over the algorithm processes of the HSC-MET scheme in detail. section 5 describes our scheme’s correct analyses. Our scheme’s securities were proven in section 6. Section 7 then compares the performance of our scheme to existing similar schemes in terms of efficiency and function. Finally, in section 8, we summarize the paper’s conclusion.

Related work
Table 1 summarizes the functional properties, confidentiality, and unforgeability analyses of the references [13–28].

The concept of heterogeneous signcryption (HSC) was proposed by Sun and Li [13]. Although their scheme realizes the heterogeneous communication from PKI to IBC, it has low security performances and high computational costs. Inspired by Sun and Li, many scholars have studied HSC. Eltayeeb et al. [14] proposed a HSC scheme without pairing computation, which realizes secure communication from CLC to PKI. Ali et al. [15] designed a HSC scheme from IBC to PKI to realize heterogeneous communication between vehicles and other entities in VANETs. The scheme supported the receivers to decrypt messages in batches, which greatly reduced the computational cost. Qiu et al. [16] proposed a HSC scheme based on the dense communication and heterogeneity of the intelligent mobile Internet of Things, which realized secure communication from IBC to CLC. The proposed scheme does not need to perform bilinear pairing operations and outsources part of the verification operations to the gateway, which greatly reduces the calculation and communication overhead of the sender and the receiver. Cao et al. [17] proposed an improved mutual HSC scheme between PKI and IBC for problems, in which the scheme of Wang et al. [18] could not resist attacks. They analyzed the security of the proposed scheme based on the assumption of the CDH problem. However, the scheme of Cao et al. uses bilinear pairing, which has a significant computational overhead. Luo et al. [19] proposed a mutual HSC scheme based on different system parameters for 5G network slices, which realized the mutual communication between CLC and PKI cryptosystem and satisfied the anonymity of messages. Ji et al. [20] proposed a mutual HSC scheme based on PKI and IBC, and proved the confidentiality and unforgeability of the scheme based on the q-Diffie-Hellman inverse problem.
The concept of PKE-ET was proposed by Yang et al. [22]. A tester can determine whether the underlying plaintext corresponding to two ciphertexts encrypted with different public keys is equal according to [22]. It has attracted the attention of many scholars. Rashad et al. [23] proposed CL-PKC-ET, a certificateless public key cryptography with equality test, to support the ciphertext equality test in IoV. Li et al. [24] designed a cryptographic scheme in IoT-based healthcare systems using proxy re-encryption and ciphertext equality test technology. The scheme realizes the flexible sharing of medical data. Shen et al. [29] proposed a group public key encryption scheme supporting equality test without bilinear pairings, G-PKEET, which greatly reduces the computational overhead. In [22–24, 29], anyone can perform the equality test algorithm on two ciphertexts, which brings many security risks. Therefore, there are many authorized equality test schemes were presented [30–32], in which only the authorized tester is promised to execute the equality test algorithm. Furthermore, some equality test schemes for heterogeneous systems have been proposed. Xiong et al. [25] proposed a HSC scheme supporting the ciphertext equality test for Internet of Drones. The concept of HSC schemes was proposed by Boneh et al. [21]. A tester can determine whether the underlying plaintext corresponding to two ciphertexts encrypted with different public keys is equal according to [21]. It has attracted the attention of many scholars. Boneh et al. [21] proposed HSCIP-ET, a certificateless public key cryptography with equality test, to support the ciphertext equality test in heterogeneous systems. They also prove the security of the scheme in ROM. Xiong et al. [26] proposed a HSC scheme from IBC to PKI with equality test (HSCIP-ET), which realized secure communication between sensors and data users. According to the IoT application scenario, Hou et al. [27] proposed a HSC scheme supporting the ciphertext equality test, which realized secure communication between sensors and data users. In [25–27], the schemes only support equality testing after grouping two ciphertexts and have many bilinear pairing operations. As a result, they face the challenges of low retrieval efficiency and high computational costs. Susilo et al.

---

Table 1. Characteristics of various works.

| Schemes       | ET | MET | SC | HSC | Without bilinear pairing | Cryptosystem | Confidentiality | Unforgeability |
|---------------|----|-----|----|-----|----------------------------|--------------|-----------------|----------------|
| Sun et al. [13] | ×  | ×   | ✓  | ✓   | ×                          | PKI—IbC      | IND-CCA         | EUF-CMA        |
| Elkhalil et al. [14] | ×  | ×   | ✓  | ✓   | ✓                          | CLC—PKI      | IND-CCA2        | EUF-CMA        |
| Ali et al. [15]   | ×  | ×   | ✓  | ✓   | ×                          | IBC—PKI      | IND-CCA2        | EUF-CMA        |
| Qiu et al. [16]   | ×  | ×   | ✓  | ✓   | ✓                          | PKI—CLC      | IND-CCA2        | EUF-CMA        |
| Cao et al. [17]   | ×  | ×   | ✓  | ✓   | ×                          | PKI—IbC      | IND-CCA2        | EUF-CMA        |
| Wang et al. [18]  | ×  | ×   | ✓  | ✓   | ×                          | PKI—IbC      | IND-CCA2        | EUF-CMA        |
| Luo et al. [19]   | ×  | ×   | ✓  | ✓   | ✓                          | PKI—CLC      | IND-CCA2        | EUF-CMA        |
| Ji et al. [20]    | ×  | ×   | ✓  | ✓   | ×                          | PKI—IbC      | IND-CCA2        | EUF-CMA        |
| Boneh et al. [21] | ×  | ×   | ×  | ×   | ×                          | PKE-KS       | IND-CCA         | ×              |
| Yang et al. [22]  | ✓  | ×   | ×  | ×   | ×                          | Public key encryption | IND-CCA         | ×              |
| Rashad et al. [23] | ✓  | ×   | ×  | ×   | ×                          | CLC encryption | IND-CCA         | ×              |
| Li et al. [24]    | ✓  | ×   | ×  | ×   | ✓                          | Proxy re-encryption | IND-CCA         | ×              |
| Xiong et al. [25] | ✓  | ×   | ✓  | ✓   | ×                          | PKI—IbC      | IND-CCA2        | EUF-CMA        |
| Xiong et al. [26] | ✓  | ×   | ✓  | ✓   | ×                          | IBC—PKI      | IND-CCA2        | EUF-CMA        |
| Hou et al. [27]   | ✓  | ×   | ✓  | ✓   | ×                          | PKI—CLC      | IND-CCA2        | EUF-CMA        |
| Susilo et al. [28] | ✓  | ✓   | ×  | ×   | ✓                          | Public key encryption | IND-CPA         | ×              |

×: not supported; ✓: supported.
IND-CCA: Indistinguishability against Chosen Ciphertext Attack.
IND-CCA2: Indistinguishability against Adaptive Chosen Ciphertext Attack.
IND-CPA: Indistinguishability against Chosen Plaintext Attack.
EUF-CMA: Existential Unforgeability against Chosen Message Attack.

https://doi.org/10.1371/journal.pone.0274695.1001
[28] proposed public-key encryption with flexible multi-ciphertext equality test (PKE-FMET) to achieve efficient ciphertext retrieval in multi-user scenarios.

Preliminaries

Complexity assumption

**Definition 1.** Computational Diffie-Hellman (CDH) problem [27]: Given a group \( G \), and \((P, aP, bP) \in G\), computing \( abP \in G \), where \( a, b \in \mathbb{Z}_q^* \).

Cramér's rule

For the non-homogeneous linear equation set

\[
\begin{align*}
    a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n &= b_1 \\
    a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n &= b_2 \\
    \vdots & \\
    a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n &= b_n
\end{align*}
\]

its coefficient determinant is

\[
\det(V) = \begin{vmatrix}
    a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
    a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{n,1} & a_{n,2} & \cdots & a_{n,n}
\end{vmatrix}.
\]

If \( \det(V) \neq 0 \), then there is a unique solution for the equation set.

Vandermonde determinant

The matrix of the form

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

called the Vandermonde matrix, and the corresponding Vandermonde determinant is

\[
\det(V) = \begin{vmatrix}
    1 & a_1 & a_1^2 & \cdots & a_1^{n-1} \\
    1 & a_2 & a_2^2 & \cdots & a_2^{n-1} \\
    \vdots & \vdots & \ddots & \vdots & \vdots \\
    1 & a_n & a_n^2 & \cdots & a_n^{n-1}
\end{vmatrix} = \prod_{1 \leq i < j \leq n} (a_i - a_j).
\]

Formal definition

The HSC-MET scheme consists of the following algorithms.

1. **Setup:** Input the system security parameter \( \lambda \), and the key generation center (KGC) and certificate authority (CA) output the system master key \( s \) and system parameter \( para \). The KGC publicizes \( para \) and keeps \( s \) secretly.

2. **PKI-Gen:** Input the identity \( ID_p \) of the PKI system user, and the CA outputs a digital certificate.

3. **CLC-PGen:** Input the system parameter \( para \) and identity \( ID_c \) of the CLC system user, and the KGC outputs the partial private-public key pair.
4. CLC-SSV: \( ID_s \) selects \( s_2 \in Z_q \) randomly and sets it as a secret value.

5. CLC-CGen: Input the system parameter \( para \), the secret value \( s_2 \), partial private-public key pair \( (SK_{c1}, PK_{c1}) \), and the user outputs the complete private-public key pair \( (SK_c, PK_c) \).

6. Trapdoor: Input the private key \( SK_c \), and the user outputs \( td_i \) as trapdoor.

7. Signcryption: Input the system parameter \( para \), the plaintext message \( m \), the receiver’s public key \( PK_r \), and the sender’s private key \( SK_p \), and the sender calculates the ciphertext \( \delta \).

8. Unsigncryption: Input the system parameter \( para \), ciphertext \( \delta \), receiver’s private key key \( SK_r \), and sender’s public key \( PK_p \), and the receiver outputs the plaintext message \( m \) or error symbol \( \perp \).

9. Test: Input ciphertexts \( \delta_i \) and trapdoors \( td_i \) where \( i \in \{1, 2, \cdots, t\} \), the cloud server outputs error symbol \( \perp \) or multi-ciphertext equality test result 0/1.

**Security model**

In the ROM, the HSC-MET scheme needs to meet the confidentiality of the message, IND-CCA2, and the unforgeability of ciphertext, EUF-CMA.

- **Confidentiality.** We define two types of adversaries, Type-1 and Type-2. A Type-1 adversary \( A_1 \) does not know the system master key, but can replace any user’s public key. A Type-2 adversary \( A_2 \) can obtain the system master key, but cannot replace any user’s public key.

**Definition 2.** If no Type-1 adversary \( A_1 \) wins game 1 with a non-negligible advantage in PPT, the HSC-MET scheme satisfies IND-CCA2–1.

**Game 1.** The game process between challenger \( C \) and adversary \( A_1 \) is as follows.

- **Setup:** \( C \) executes the setup algorithm, outputs the system parameter \( para \) and the master key \( s \), returns \( para \) to \( A_1 \), and stores \( s \) secretly.

- **Phase 1:** \( A_1 \) can perform limited following polynomial queries.

  - **Partial private key query:** \( A_1 \) queries for the partial private key of \( ID_i \), \( C \) executes the CLC-PGen algorithm to generate \( SK_{c1} \) and return it to \( A_1 \).

  - **Private key query:** \( A_1 \) queries for the private key of \( ID_i \), \( C \) executes the CLC-CGen algorithm to generate \( (SK_c, PK_c) \) and return \( SK_c \) to \( A_1 \).

  - **Public key query:** \( A_1 \) queries for the public key of \( ID_i \), \( C \) executes the CLC-CGen algorithm to generate \( (SK_c, PK_c) \) and return \( PK_c \) to \( A_1 \).

  - **Replace public key query:** \( A_1 \) can select any public key \( PK_{c2} \) to replace the original public key \( PK_c \).

  - **Trapdoor query:** \( A_1 \) queries for the trapdoor of \( ID_c \), \( C \) executes the Trapdoor algorithm to generate \( td_i \) and return it to \( A_1 \).

  - **Signcryption query:** When receiving the query with \( (m_i, ID_{pi}, ID_{ci}) \) submitted by \( A_1 \), \( C \) executes the Signcryption algorithm to generate \( \delta_i \), and returns it to \( A_1 \).

  - **Unsigncryption query:** When receiving the query with \( (ID_{pi}, ID_{ci}, \delta_i) \) submitted by \( A_1 \), \( C \) executes the Unsigncryption algorithm to generate \( m_i \), and returns it to \( A_1 \).

- **Challenge:** \( A_1 \) selects the sender’s identity \( ID_{p} \), receiver’s identity \( ID_{r} \) and two plaintexts of equal length \( m_0 \) and \( m_1 \) to \( C \). \( C \) selects randomly \( \xi \in [0, 1] \) and performs the signcryption algorithm to generate ciphertext \( \delta^* \) and return it to \( A_1 \).
**Phase 2:** After receiving $\delta^*$, the adversary $A_i$ continues to execute the queries in Phase 1. However, $A_i$ can neither query the private key of $ID^*_p$, nor can $A_i$ make unsigncryption query of $(\delta^*, ID^*_p, ID^*_r)$. $A_i$ also can’t query the trapdoor of $ID^*_r$.

**Guess:** $A_i$ outputs a guess value $\xi \in \{0, 1\}$. $A_i$ wins the game if $\xi^* = \xi$. We define the advantage of $A_i$ as $Adv_{A_i}^{\text{IND-CCA}2-1}(\lambda) = |Pr[\xi^* = \xi] - \frac{1}{2}|$, where $Pr[\xi^* = \xi]$ represents the probability of $\xi^* = \xi$.

**Definition 3.** If no Type-2 adversary $A_2$ wins game 2 with a non-negligible advantage in PPT, the HSC-MET scheme satisfies IND-CCA2–2 security.

**Game 2.** The game process between challenger $C$ and adversary $A_2$ is as follows.

**Setup:** $C$ executes the setup algorithm, outputs the system parameter $para$ and the master key $s$, and returns them to $A_2$.

**Phase 1:** $A_i$ can perform all the queries in Definition 2 except the replace public key query. The challenge, phase 2, and guess stage are the same as Definition 2 and will not be repeated here. We define the advantage of $A_2$ as $Adv_{A_2}^{\text{IND-CCA}2-2}(\lambda) = |Pr[\xi^* = \xi] - \frac{1}{2}|$, where $Pr[\xi^* = \xi]$ represents the probability of $\xi^* = \xi$.

**Unforgeability. Definition 4.** If no adversary $F$ wins Game 3 with a non-negligible advantage $e$ in PPT, the HSC-MET scheme can satisfy EUF-CMA security.

**Game 3.** The game between challenger $C$ and adversary $F$ is as follows.

**Training:** $F$ can perform limited following polynomial queries.

- **Key query:** $F$ queries for the public key of $ID_p$ and $C$ executes the PKI-Gen algorithm to generate $(SK_p, PK_p)$ and return to $F$.
- **Signcryption query:** When receiving the query with $(m, ID_p, ID_c)$ submitted by $F$, $C$ executes the signcryption algorithm to generate $d_i$ and returns it to $F$.
- **Unsigncryption query:** When receiving the query with $(ID_p, ID_c, \delta)$ submitted by $F$, $C$ executes the unsigncryption algorithm to obtain $m_i$ and returns it to $F$.
- **Forgery:** $F$ selects the sender’s identity $ID^*_p$ and the receiver’s identity $ID^*_r$, and forges a ciphertext $\delta^*$. If $\delta^*$ can meet the following requirements, $F$ can win the Game 3.

The error symbol $\bot$ will not be returned when the unsigncryption query is performed on $(\delta^*, ID^*_p, ID^*_r)$.

The adversary $F$ can not query for the private key $SK^*_p$ of the $ID^*_p$.

$\delta^*$ cannot be generated by the signcryption query of $(m^*, ID^*_p, ID^*_r)$.

We define the advantage of $F$ to win in this game as $Adv_{F}^{\text{EUF-CMA}}(\lambda) = Pr[F\text{ wins}].$

**Scheme design**

1. **Research questions and methodologies:** Table 2 displays the main research problems and relevant solutions of this paper, which are based on the previous relevant work subsection’s collections and analyses of references.

2. **Scheme processes:**

- **Setup:** The KGC and CA initialize the system and generate the system parameters.
- **User-Gen:** The CA generates digital certificates for UAVs in PKI. The KGC generates partial keys for data users in CLC.
• **Signcrypt and upload:** UAVs signcrypt the collected data and upload it to the cloud server.

• **Test:** The cloud server performs the equality test for multi-ciphertexts.

• **Download and unsigncrypt:** Data users download and unsigncrypt data from the cloud server.

3. **System model:** The system model of our scheme is composed of five entities: KGC, CA, UAVs, cloud server and data users. The functions of each entity are as follows. The system model diagram is shown in Fig 1.

   - **KGC.** The KGC initializes the system, generates the key and system parameter, and distributes partial keys to data users.

   - **CA.** The CA issues digital certificates for UAVs.

   - **UAVs.** UAVs collect and signcrypt the collected environmental data, and upload it to the cloud server.

   - **Cloud server.** The cloud server stores the uploaded ciphertext, and processes the data user’s request to execute the test algorithm, and returns the test result to the users.

   - **Data users.** Users who wish to obtain environmental data, such as monitoring personnel and data processing centers, are responsible for submitting the trapdoor of the ciphertext equality test to the cloud server and verifying the ciphertext that meets the requirements.

| No. | Problem                                                                 | Solution                                           |
|-----|-------------------------------------------------------------------------|----------------------------------------------------|
| 1   | How can data confidentiality and unforgeability be achieved at the same time? | Signcryption                                       |
| 2   | How to realize the secure communication between heterogeneous entities in IoT? | Heterogeneous signcryption                          |
| 3   | How to achieve secure storage of data on cloud servers?                 | Store ciphertext                                    |
| 4   | How to efficiently retrieve the ciphertexts on the cloud server?        | Multi-ciphertext equality test                     |
| 5   | How to reduce the computational cost of the scheme?                     | Without bilinear pairing                            |

https://doi.org/10.1371/journal.pone.0274695.t002
Our construction

1. **Setup**: Given the system security parameter $\lambda$, KGC selects a large prime number $q (q \geq 2^\lambda)$ and an additive cyclic group $G$ with order $q$ and generator $P$. Four hash functions, $H_1: \{0,1\}^* \rightarrow Z_q^*$, $H_2: G \rightarrow \{0,1\}^{2\lambda}$, $H_3: G \rightarrow \{0,1\}^n$ and $H_4: \{0,1\}^* \rightarrow \{0,1\}^n$ are defined. KGC randomly selects $s \in Z_q^*$ as the system master key $SK$ and calculates the public key $PK = sp$. It also selects the maximum number of ciphertexts that can perform the multi-ciphertext equality test, $n$. KGC sets and exposes $para = \{\lambda, G, q, P, PK, H_1, H_2, H_3, H_4, n\}$.

2. **PKI-Gen**: $ID_p$ selects $s_p \in Z_q^*$ randomly, and calculates $PK_p = s_pP$. The user sends $(ID_p, PK_p)$ to CA which generates a digital certificate for it.

3. **CLC-PGen**: When receiving the registration request from $ID_o$, KGC randomly selects $s_1 \in Z_q^*$, and calculates $PK_{c_1} = s_1P$ and $SK_{c_1} = s_1 + SKH_1(ID_o)$. Then the KGC returns $(SK_{c_1}, PK_{c_1})$ to $ID_c$ securely.

4. **CLC-SSV**: $ID_x$ randomly selects $s_2 \in Z_q^*$ as a secret value.

5. **CLC-CGen**: $ID_x$ sets $SK_{c_2} = s_2P$, $PK_{c_2} = s_2P$, $SK_c = (SK_{c_1}, SK_{c_2})$ and $PK_c = (PK_{c_1}, PK_{c_2})$.

6. **Trapdoor**: Input the private key $SK_c = (SK_{c_1}, SK_{c_2})$, output the $td_c = SK_{c_2}$.

7. **Signcryption**: Input $(para, m, PK_c, SK_c)$ and output $\delta$. Specific steps are as follows.
   a. Calculate $f_{0,n} = H_1(m||n)$ and $f_{i,n} = H_1(m||n||f_{0,n}||\cdots||f_{i-1,n})$ where $i \in \{1, 2, \cdots, n - 1\}$.
   b. Calculate $f_{i,j} = H_1(f_{i-1,j})$ where $i \in \{k, \cdots, n - 1\}$ & $j \in \{0, 1, \cdots, i - 1\}$. And calculate $f_{i,x} = f_{0,i} + f_{1,i} x + \cdots + f_{i-1,i} x^{i-1}$, $i \in \{k, \cdots, n\}$, where $k$ is the number of ciphertexts that can be tested for equality.
   c. Select $r, X \in Z_q^*$ randomly. Calculate $Y = SK_p(PKH_1(ID) + PK_{c_1})$ and $R = rPK_{c_2}$.
   d. Calculate $C_1 = rP, C_2 = (m||r) \oplus H_2(Y) \oplus H_2(R), C_3 = (X||f_{k,x}||\cdots||f_{n,x}) \oplus H_3(R)$ and $C_4 = H_4(C_1||C_2||C_3||f_{0,k}||f_{1,k}||\cdots||f_{k-1,k}||R||k)$.
   e. Output the ciphertext $\delta = (C_1, C_2, C_3, C_4, k)$.

8. **Unsigncryption**: Input $(para, \delta, PK_{c'}, SK_c)$ and output $m'$ or $\bot$. Specific steps are as follows.
   a. Calculate $Y' = SK_p(PK_{c'}), R = SK_{c'}C_1$ and $m'||r' = H_2(Y') \oplus H_2(R') \oplus C_2$.
   b. Calculate $f'_{0,x} = H_1(m'||n)$ and $f'_{i,x} = H_1(m'||n||f'_{0,x}||\cdots||f'_{i-1,x})$ where $i \in \{1, 2, \cdots, n - 1\}$.
   c. Calculate $f'_{i,j} = H_1(f'_{i-1,j})$ where $i \in \{k, \cdots, n - 1\}$ & $j \in \{0, 1, \cdots, i - 1\}$ and $X'||f'_{k,x}||\cdots||f'_{n,x} = C_3 \oplus H_3(R')$.
   d. Verify that the Eqs (1), (2) and (3) are true, where $i \in \{k, \cdots, n\}$.

\[ C_1 = r'P \]  \hspace{1cm} (1)

\[ f'_{i,x} = f'_{0,i} + f'_{1,i}x + \cdots + f'_{i-1,i}x^{i-1} \]  \hspace{1cm} (2)

\[ C_1 = H_1(C_1||C_2||C_3||f'_{0,k}||f'_{1,k}||\cdots||f'_{k-1,k}||R'||k) \]  \hspace{1cm} (3)

If the equations are all true, return $m'$. Otherwise, return $\bot$.
9. **Test:** Input $t$ ciphertexts $\delta_i = (C_{i1}, C_{i2}, C_{i3}, C_{i4}, k_i)$ and trapdoors $td_i$. Let $k = \max\{k_1, k_2, \cdots, k_t\}$. If $k \leq \min\{t, n\}$, perform the following computations. Otherwise, return $\bot$.

a. Calculate $X||f_{i,k}(X)|| \cdots ||f_{n,k}(X) = C_{i3} \oplus H_3(td_{i1})$ and extract $f_{1,k}(X_1), f_{2,k}(X_2), \cdots, f_{k-1,k}(X_{k-1})$ from the ciphertexts.

b. Assume that the plaintexts corresponding to $t$ ciphertexts $\delta_i$ are equal. By calculating $f_{j,k}$, we can get the non-homogeneous linear equation set

\[
\begin{align*}
    f_{i,k}^i(X_1) &= f_{i,k}^0 + f_{i,k}^1 X_1 + f_{i,k}^2 X_1^2 + \cdots + f_{i,k}^{k-1} X_1^{k-1} \\
    f_{i,k}^j(X_2) &= f_{i,k}^0 + f_{i,k}^1 X_2 + f_{i,k}^2 X_2^2 + \cdots + f_{i,k}^{k-1} X_2^{k-1} \\
    &\vdots \\
    f_{i,k}^k(X_{k}) &= f_{i,k}^0 + f_{i,k}^1 X_k + f_{i,k}^2 X_k^2 + \cdots + f_{i,k}^{k-1} X_k^{k-1}
\end{align*}
\]

Let $f_{i,k}^i = f_{j,k}^i$ where $i_1, i_2 \in \{1, 2, \cdots, k\}$ and $j \in \{1, 2, \cdots, k - 1\}$. If $f_{i,k}^0, f_{k-1,k}^0$ is regarded as the solution of the equation set. And $X_i$ is regarded as the coefficient. $\det(V) \neq 0$ can be known according to Kramer’s rule and Vandermonde determinant. The unique solution $f_{0,k}^i, f_{i,k}^1, \cdots, f_{k-1,k}^1$ of the equation set can be obtained.

c. For each ciphertext $\delta_i = (C_{i1}, C_{i2}, C_{i3}, C_{i4}, k_i)$, verify whether the equation $C_{i4} = H_4(C_{i1} || C_{i2} || f_{0,k}^i || f_{1,k}^i || \cdots || f_{k-1,k}^i || td_{i1} || |k|)$ holds. If the equation is true for every $\delta_i$, it represents $m_1 = m_2 = \cdots = m_i$, and the test result 1 is returned. Otherwise, 0 is returned.

### Correctness analysis

**Theorem 1.** The unsigncryption algorithm is correct.

**Proof.** The correctness of the unsigncryption algorithm can be verified by the following two equations.

1. After receiving the ciphertext $\delta = (C_{11}, C_{12}, C_{13}, C_{14}, k_1)$, the data user can get $m'$ by calculating $m' || r' = H_2(Y') \oplus H_2(R') \oplus C_2$. Eq (4) holds.

\[
m' || r' = H_2(Y') \oplus H_2(R') \oplus C_2 = H_2(SK_{C_{12}} PK_1) \oplus H_2(SK_{C_{14}} C_1) \oplus C_2
\]

2. The data user can calculate $X||f_{i,k}(X)|| \cdots ||f_{n,k}(X) = C_{i3} \oplus H_3(SK_{C_{12}} C_1)$ to verify the legitimacy of the message and signature. Eq (5) holds.

\[
X||f_{i,k}(X)|| \cdots ||f_{n,k}(X) = C_{i3} \oplus H_3(R')
\]

\[
= (X || f_{i,k}(X)|| \cdots || f_{n,k}(X)) \oplus H_3(rPK_{C_{12}}) \oplus H_3(SK_{C_{14}} C_1)
\]

\[
= (X || f_{i,k}(X)|| \cdots || f_{n,k}(X)) \oplus H_3(rSK_{C_{14}} P) \oplus H_3(SK_{C_{12}} P)
\]

\[
= X || f_{i,k}(X)|| \cdots || f_{n,k}(X)
\]

Through the above verification, theorem 1 is established.

**Theorem 2.** The Test algorithm is correct.
Proof. The correctness of the Test algorithm can be verified by the following equations. Given $t$ ciphertexts $\delta = (C_{i1}, C_{i2}, C_{i3},\ldots, C_{it})$, let $k = \max\{k_1, k_2, \ldots, k_t\}$. If $k \leq \min\{t, n\}$, calculate $X_i = ||f_{ik}(X_i)|| \cdots ||f_{in}(X_i)|| = C_{i1} \oplus H_t(td/C_{i1})$.

Assume that the plaintexts of $t$ ciphertexts $\delta_i$ are $m_1, m_2, \ldots, m_t$ respectively.

1. When the plaintexts corresponding to the tested $t$ ciphertexts are equal, the correctness of the Test algorithm is proved as follows. If $m_1 = m_2 = \cdots = m_t$, we must have $f_{ik}^1 = f_{ik}^j$ where $i_1, i_2 \in \{1, 2, \ldots, t\}$ and $j \in \{0, 2, \ldots, t - 1\}$. Let $f_{ik}^j = f_{jk}^1$. We can get the equation set Eq (6).

$$
\begin{align*}
\left\{\begin{array}{l}
f_{ik}^1(X_1) &= f_{0k}^1 + f_{1k}^1X_1 + \cdots + f_{k-1,k}^1X_k^{k-1} \\
f_{ik}^1(X_2) &= f_{0k}^1 + f_{1k}^1X_2 + \cdots + f_{k-1,k}^1X_k^{k-1} \\
& \vdots \nonumber \\
f_{ik}^1(X_k) &= f_{0k}^1 + f_{1k}^1X_k + \cdots + f_{k-1,k}^1X_k^{k-1}
\end{array}\right.
\end{align*}
$$

(6)

If $f_{0k}^1, f_{1k}^1, \cdots, f_{k-1,k}^1$ is regarded as the solution of the equation set, and $X_i$ is regarded as a coefficient. The equation set corresponds to the Vandermonde matrix Eq (7).

$$
V = \begin{bmatrix}
X_1 & X_1^2 & \cdots & X_1^{k-1} \\
X_2 & X_2^2 & \cdots & X_2^{k-1} \\
\vdots & \vdots & \ddots & \vdots \\
X_k & X_k^2 & \cdots & X_k^{k-1}
\end{bmatrix}
$$

(7)

The determinant of $V$ is $\det(V) = \prod_{1 \leq i < j \leq k} (X_i - X_j)$. Due to the randomness of $X_i$, the probability of $\det(V) = 0$ is $\frac{1}{(k-1)!}$. The equation set has a unique solution $f_{0k}^1, f_{1k}^1, \cdots, f_{k-1,k}^1$ when $\det(V) \neq 0$ from Cramer’s rule. For each ciphertext $\delta_i$, the equation $C_{i1} = H_t(C_{i1} || C_{i2} || C_{i3} || f_{0k}^1 || f_{1k}^1 || \cdots || f_{k-1,k}^1 || td/C_{i1}) || k_i)$ holds. We can get $Test(para, \delta_1, \delta_2, \cdots, \delta_k, td_1, td_2, \cdots, td_k) = 1$.

2. When the plaintexts corresponding to the tested $t$ ciphertexts are not equal, the correctness of the Test algorithm is proved as follows. If $m_1 \neq m_2 = \cdots = m_t$, there is $f_{ik}^1 \neq f_{ik}^j$ where $i_1, i_2 \in \{2, 3, \ldots, t\}$ and $j \in \{0, 2, \ldots, t - 1\}$. We can obtain the equation set Eq (8).

$$
\begin{align*}
\left\{\begin{array}{l}
f_{ik}^1(X_1) &= f_{0k}^1 + f_{1k}^1X_1 + \cdots + f_{k-1,k}^1X_k^{k-1} \\
f_{ik}^1(X_2) &= f_{0k}^1 + f_{1k}^1X_2 + \cdots + f_{k-1,k}^1X_k^{k-1} \\
& \vdots \nonumber \\
f_{ik}^1(X_k) &= f_{0k}^1 + f_{1k}^1X_k + \cdots + f_{k-1,k}^1X_k^{k-1}
\end{array}\right.
\end{align*}
$$

(8)

Let $f_{ik}^j = f_{jk}^1$ where $j \in \{0, 2, \ldots, t - 1\}$. The unique solution $f_{0k}^1*, f_{1k}^1*, \cdots, f_{k-1,k}^1*$ can be obtained. It cannot make the Eqs (9) and (10) hold at the same time.

$$
\begin{align*}
C_{1,k} &= H_t(C_{11} || C_{21} || C_{13} || f_{0k}^1* || f_{1k}^1* || \cdots || f_{k-1,k}^1* || td/C_{11}) || k_i) \\
C_{2,k} &= H_t(C_{21} || C_{22} || C_{23} || f_{0k}^1* || f_{1k}^1* || f_{2k}^1* || \cdots || f_{k-1,k}^2* || td/C_{22}) || k_i)
\end{align*}
$$

(9)

(10)

Through the above verification, theorem 2 is established.
Security proofs

Confidentiality

Theorem 3. If an adversary \( \mathcal{A}_1 \) can win the Game 1 in PPT with a non-negligible advantage \( \varepsilon_1 \) after \( q_h(i = 1, 2, 3, 4) \) \( H_i \) queries, \( q_d \) partial private key queries, \( q_{sc} \) signcryption queries and \( q_{usc} \) unsigncryption queries, the challenger \( C \) can solve the CDH problem with the nonnegligible advantage \( \varepsilon_1' \) as show in Eq (11).

\[
\varepsilon_1' = \left(1 - \frac{4d}{q_h}\right) \left(1 - \frac{q_d(q_h + q_{sc} + q_{usc})}{2^\lambda}\right) \left(1 - \frac{q_{usc}}{2^\lambda}\right) \varepsilon_1
\] (11)

Proof: \( C \) is a challenger to solve the CDH problem. \( \mathcal{A}_1 \) is a Type-1 adversary. Given a challenge example \((P, aP, bP)\) where \( a, b \in \mathbb{Z}_q \), \( C \) and \( \mathcal{A}_1 \) interact as follows.

Setup: \( C \) executes the setup algorithm to output the system parameter \( para = \{\lambda, G, q, P, PK, H_1, H_2, H_3, H_4, n\} \).

Phase 1: \( C \) needs to maintain initially empty lists \( L_{hi}, i = 1, 2, 3, 4, L_d, L_{sk}, L_{pk} \) and \( L_{td} \) to record the query results of \( A_1 \).

- \( H_1 \) query: When receiving the query with \( ID, h_1 \) submitted by \( A_1 \), \( C \) searches for whether there is \((ID, h_1)\) in \( L_{h1} \). When it exists, \( C \) returns \( h_1 \) to \( A_1 \). Otherwise, \( C \) selects \( h_1 \in \mathbb{Z}_q^* \) randomly and returns to \( A_1 \). \( C \) inserts \((ID, h_1)\) into \( L_{h1} \) finally.

- \( H_2 \) query: When receiving the query with \( R, h_2 \) submitted by \( A_1 \), \( C \) searches for whether there is \((R, h_2)\) in \( L_{h2} \). When it exists, \( C \) returns \( h_2 \) to \( A_1 \). Otherwise, \( C \) randomly selects \( h_2 = \{0, 1\}^q \) and returns to \( A_1 \). And \( C \) inserts \((R, h_2)\) into \( L_{h2} \).

- \( H_3 \) query: When receiving the query with \( r_n , PK_{ic2} \) submitted by \( A_1 \), \( C \) searches for whether there is \((r_n , PK_{ic2}, h_3)\) in \( L_{h3} \). When it exists, \( C \) returns \( h_3 \) to \( A_1 \). Otherwise, \( C \) randomly selects \( h_3 = \{0, 1\}^q \) and returns to \( A_1 \). And \( C \) inserts \((r_n , PK_{ic2}, h_3)\) into \( L_{h3} \).

- \( H_4 \) query: When receiving the query with \( (C_{1,i}, C_{2,i}, C_{3,i}, f_{ik}, r_n , PK_{ic2}, k_i) \) submitted by \( A_1 \), \( C \) searches for whether there is the corresponding \( h_i \) in \( L_{h4} \). When it exists, \( C \) returns \( h_i \) to \( A_1 \). Otherwise, \( C \) selects \( h_i = \{0, 1\}^q \) randomly and returns to \( A_1 \). And \( C \) inserts \((C_{1,i}, C_{2,i}, C_{3,i}, f_{ik}, k_i, r_n, PK_{ic2}, h_i)\) into \( L_{h4} \).

- Partial private key query: When receiving the query with \( ID_{ci} \) from \( A_1 \), if \((ID_{ci}, SK_{ic1})\) exists, \( C \) returns it to \( A_1 \). Otherwise, \( C \) executes CLC-PGen algorithm to generate \( SK_{ic1} \) and return to \( A_1 \). \( C \) inserts \((ID_{ci}, SK_{ic1})\) into \( L_{d} \).

- Private key query: When receiving the query with \( ID_{ci} \) from \( A_1 \), if \((ID_{ci}, SK_{ci})\) exists, \( C \) returns it to \( A_1 \). Otherwise, \( C \) executes CLC-CGen algorithm to generate \( SK_{ci} \) and return to \( A_1 \). \( C \) inserts \((ID_{ci}, SK_{ci})\) into \( L_{sk} \).

- Public key query: When receiving the query with \( ID_{ci} \) from \( A_1 \), if \((ID_{ci}, PK_{ci})\) exists, \( C \) returns it to \( A_1 \). Otherwise, \( C \) executes CLC-PGen algorithm to generate \( PK_{ci} \) and return to \( A_1 \). \( C \) inserts \((ID_{ci}, PK_{ci})\) into \( L_{pk} \).

- Replace public key query: \( A_1 \) can select any public key \( PK_{ci}^* \) to replace the user’s original public key \( PK_{ci} \).

- Trapdoor query: When receiving the query with \( ID_{ci} \) from \( A_1 \), if \((ID_{ci}, td_{ci})\) exists, \( C \) returns it to \( A_1 \). Otherwise, \( C \) executes Trapdoor algorithm to generate \( td_{ci} \) and return to \( A_1 \), and inserts \((ID_{ci}, td_{ci})\) into \( L_{td} \).
Theorem 5. Unforgeability

Unsigncryption query: When receiving the query with \((m_i, ID_{p_i}, ID_{c_i})\) submitted by \(A_i\), \(C\) executes the Unsigncryption algorithm to obtain the plaintext \(m_i\) and returns it to \(A_i\).

Challenge: \(A_i\) submits the sender’s identity \(ID_{p_i}^*\), receiver’s identity \(ID_{c_i}^*\), and two plaintexts \(m_0\) and \(m_1\) of the same length to \(C\). \(A_i\) has never asked for the private key for \(ID_{c_i}^*\). \(C\) randomly selects \(a \in Z_q^*\) as the secret value of \(ID_{c_i}^*\) and calculates \(PK_{c_i}^* = aP\). Then \(C\) randomly selects \(\xi \in \{0, 1\}\) and performs the following calculations.

- Calculate \(f_{0,n} = H_i(m_d||n)\) and \(f_{1,n} = H_i(m_d||\xi||f_{0,n}||\cdots||f_{i-1,n})\) where \(i \in \{1, 2, \ldots, n-1\}\).
- Calculate \(f_{j,i} = H_i(f_{j,i+1})\) where \(i \in \{k, k+1, \ldots, n-1\}\) and \(j \in \{0, 1, \ldots, i-1\}\).
- Calculate \(f_i(x) = f_{0,i} + f_{1,i} x + \cdots + f_{1,i-1} x^{i-1}\) where \(i \in \{k, k+1, \ldots, n\}\).
- Randomly select \(b, X \in Z_q^*\), and calculate \(Y^* = SK_{c_i}^*(PK_{c_i}^*(ID_{c_i}^* + PK_{c_i}^*) + PK_{c_i}^*\) and \(R^* = bPK_{c_i}^*\).
- Calculate \(C_i^* = bP, C_i^* = (m_i||b) \oplus H_i(Y^* \oplus H_i(R^*)), C_i^* = (X||f_i(X)||\cdots||f_{1,i} X||H_3(R^*))\) and \(C_i = H_i(C_i^*)||C_i^*||f_{0,i}||f_{1,i}||\cdots||f_{1,i-1}||R^*||k\).
- \(C\) returns \(\delta^* = (C_1, C_2, C_3, C_4, k)\) to \(A_i\).

Phase 2: \(A_i\) continues to perform the queries after receiving \(\delta^*\), but \(A_i\) cannot query the private key of the \(ID_{c_i}^*\) nor can it perform unsigncryption query on \(\delta^*\).

Guess: \(A_i\) outputs the guess value \(\xi^*\). If \(\xi^* = \xi, A_i\) wins the game. \(C\) will select \((R_0, H_2(R_0))\) from the list \(L_{h_2}\) and take \(R_1 = abP\) as the solution of the CDH problem. However, there is currently no effective way to solve the CDH problem. Theorem 3 is proved.

Theorem 4. If an adversary \(A_i\) can win the Game 2 in PPT with a non-negligible advantage \(\varepsilon_2\) after \(q_{h_i}(i = 1, 2, 3, 4) H_i\) queries, \(q_d\) partial private key queries, \(q_{sc}\) signcryption queries and \(q_{usc}\) unsigncryption queries, the challenger \(C\) can solve the CDH problem with the advantage \(\varepsilon_2\) as shown in Eq (12).

\[
\varepsilon_2' = \left(1 - \frac{q_d}{q_{h_i}}\right)\left(1 - \frac{q_{usc} (q_{h_i} + q_{h_i} + q_{h_i})}{2^\xi}\right)\left(1 - \frac{q_{usc}}{2^\xi}\right)\varepsilon_2 \tag{12}
\]

The proof process is similar to Theorem 3 and will not be repeated here.

Unforgeability

Theorem 5. If an adversary \(\mathcal{F}\) can win the Game 3 in PPT with a non-negligible advantage \(\varepsilon_3\) after \(q_{h_i}(i = 1, 2, 3, 4) H_i\) queries, \(q_{pk}\) public key queries and \(q_{sc}\) signcryption queries, the challenger \(C\) can solve the CDH problem with the advantage \(\varepsilon_3\) as shown in Eq (13).

\[
\varepsilon_3' = \left(1 - \frac{q_{pk}}{2^\xi}\right)\left(1 - \frac{q_{usc} (q_{h_i} + q_{h_i} + q_{h_i})}{2^\xi}\right)\varepsilon_3 \tag{13}
\]

Proof: \(C\) is a challenger to solve the difficult problems of CDH. \(\mathcal{F}\) is an adversary. \(C\) selects \(ID_{p_i}^*\) as the challenge identity. Given a challenge example \((P, aP, bP)\) where \(a, b \in Z_q^*\), \(C\) and \(\mathcal{F}\) interact as follows.

Setup: \(C\) randomly selects \(a \in Z_q^*\) and calculates \(PK = aP\). \(C\) outputs the system parameter \(para = \{\lambda, G, q, P, PK, H_1, H_2, H_3, H_4, n\}\).
In this section, we proved the number security of our scheme based on the definition of number security in reference [28].

**Key query:** When receiving the query with \((ID_p, ID_c, m_i)\) submitted by \(\mathcal{F}\), \(\mathcal{C}\) executes the PKI-Gen algorithm to generate \((SK_p, PK_p)\) and return to \(\mathcal{F}\) if \(ID_p \neq ID'_p\). Otherwise, \(\mathcal{C}\) randomly selects \(b \in Z_q^*\) and calculates \(PK_p = bP\). Then \(\mathcal{C}\) returns \(PK_p\) to \(\mathcal{F}\).

**Signcryption query:** When receiving the query with \((ID_p, ID_c, m_i)\) submitted by \(\mathcal{F}\), \(\mathcal{C}\) executes the signcryption scheme to generate \(\delta'_i\) and return to \(\mathcal{F}\) if \(ID_p \neq ID'_p\). Otherwise, \(\mathcal{C}\) performs the following operations.

- Calculate \(f_{0,n} = H_1(m||n)\) and \(f_{i,n} = H_1(m||n||f_{0,n}||\cdots||f_{i-1,n})\) where \(i \in \{1, 2, \cdots, n-1\}\).
- Calculate \(f_{i} = H_1(f_{i+1})\) where \(i \in \{k, \cdots, n-1\}\) and \(j \in \{0, \cdots, i-1\}\).
- Calculate \(f_{i}(x) = f_{0,i} + f_{1,i}x + \cdots + f_{i-1,i}x^{i-1}\) where \(i \in \{k, k+1, \cdots, n\}\).
- Randomly select \(r, X' \in Z_q^*\). Calculate \(Y' = b(PKH_1(ID_c) + PK_{k+1})\) and \(R' = rPK_{k+2}\).
- Calculate \(C_i = rP, C_2 = (m||r) \oplus H_2(Y') \oplus H_3(R'), C_3 = (X||f_3(X)) \cdots ||f_n(X)) \oplus H_3(R')\) and \(C_i = H_4(C_1||C_2||C_3||f_{0,k}||f_{1,k}||\cdots||f_{k-1,k}||R'||k)\).
- Return \(\delta' = (C_1, C_2, C_3, C_4, k_i)\) to \(\mathcal{F}\).

**Forgery:** \(\mathcal{F}\) outputs a forged ciphertext \(\delta'_i = (C'_1, C'_2, C'_3, C'_4, k_i)\) for \(m_i\). If the forgery is successful, \(\mathcal{C}\) can select \((Y', H_3(Y'))\) from the list \(L_{h2}\) and take \(abP = rPK_{h2}\) as the solution of the CDH problem. However, there is currently no effective way to solve the problem. Theorem 5 is proved.

**Number security**

In this section, we proved the number security of our scheme based on the definition of number security in reference [28].

**Theorem 6.** If there is an adversary \(\mathcal{A}\), after \(q_{H_i}(i = 1, 2, 3, 4)\) \(H_i\) queries, \(q_{td}\) trapdoor queries, \(q_{sc}\) signcryption queries and \(q_{usc}\) unsigncryption queries, can determine whether the underlying plaintext corresponding to \(t < K\) ciphertexts \(\delta_i = (C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}, k_i)\) is equal in PPT with a non-negligible advantage \(\varepsilon_{\delta_i}\) where \(k = \max\{k_1, k_2, \cdots, k_t\}\). \(\mathcal{C}\) can solve the problem of CDH with the advantage \(\varepsilon'_{\delta_i}\) as show in Eq (14).

\[
\varepsilon'_{\delta_i} = \left(1 - \frac{q_{td}}{2^t}\right) \left(1 - \frac{q_{sc}(q_{td} + q_{h2} + q_{h3})}{2^t}\right) \left(1 - \frac{q_{usc}}{2^t}\right) \varepsilon_{\delta_i} \tag{14}
\]

**Proof:** There are the following two ways to determine for \(\mathcal{A}\).

1. \(\mathcal{A}\) can determine by obtaining and comparing the plaintexts \(m_1, m_2, \cdots, m_t\).
   In subsection Confidentiality, we have proved the confidentiality of our scheme. So this way is not feasible for \(\mathcal{A}\).

2. \(\mathcal{A}\) can determine by obtaining the value of \(f_{i}^d\), where \(i \in \{1, 2, \cdots, t-1\}\) and \(k_i < k' < n\). For this way, we do the following analysis.

   For \(t\) ciphertexts \(\delta_1, \delta_2, \cdots, \delta_t\), \(\mathcal{A}\) is allowed to perform public key queries and trapdoor queries. So it can calculate \(X_i||f_3(X_i)||\cdots||f_n(X_i) = C_{i,3} \oplus H_3(td_{i}C_{i,1})\) and \(f_1(X_i) = f_{0,i} + f_{1,i}X_i + \cdots + f_{i-1,i}X_i^{i-1}\).
\[ \cdots + f_{i-1} X_{i-1} \] where \( k_i < j < n \). Let \( k = \max\{k_1, \cdots, k_t\} \). \( A \) can get the equation set Eq (15).

\[
\begin{align*}
A(X_1) &= f_{0,1}^{1} + f_{1,1}^{1} X_1 + f_{2,1}^{1} X_1^2 + \cdots + f_{k-1,1}^{1} X_1^{k-1} \\
A(X_2) &= f_{0,2}^{1} + f_{1,2}^{1} X_2 + f_{2,2}^{1} X_2^2 + \cdots + f_{k-1,2}^{1} X_2^{k-1} \\
& \vdots \\
A(X_t) &= f_{0,t}^{1} + f_{1,t}^{1} X_t + f_{2,t}^{1} X_t^2 + \cdots + f_{k-1,t}^{1} X_t^{k-1}
\end{align*}
\]  

(15)

Since \( X_1, X_2, \cdots, X_t \) are randomly selected by users, the probability of non-linear correlation of \( t \) equations is \( p = \frac{2^n-1}{2^{k+1}} \). Let \( f_{i,j}^{j} \) be an independent variable, where \( i \in \{1, 2, \cdots, t\} \) and \( j \in \{0, 1, \cdots, k-1\} \). If \( f_{0,k}^{1}, f_{1,k}^{1}, \cdots, f_{k-1,k}^{1} \) is regarded as a coefficient of the equations, the equation set consisting of \( t \) equations with \( k \) independent variables can be obtained. Because of \( k > t \), there is no solution to make the equation set true. So this way is not feasible for \( A \).

In summary, the HSC-MET scheme satisfies the number security. Theorem 6 is proved.

### Scheme analysis

#### Functional analysis

Table 3 summarizes the functional properties, confidentiality, and unforgeability analyses of our scheme. can be seen in Tables 1 and 3. To begin, in comparison to the references [13–20], our scheme incorporates the MET function to achieve safe and efficient cloud data retrieval, which is more appropriate for application scenarios involving large amounts of data. Second, when compared to the schemes in [22–27], our scheme overcomes the limitation of only supporting pairwise grouping for ciphertext equality tests, making it more appropriate for multi-user and multi-ciphertext application scenarios. Third, unlike [13, 15, 17, 18, 20, 22–27], our scheme does not use bilinear pairing and has lower computational costs. Furthermore, our scheme has higher confidentiality than [13, 22–24, 27, 28]. Finally, unlike [22–24, 28], our scheme has unforgeability and introduces heterogeneous signcryption technology to ensure the confidentiality, and integrity of data, and realizes the secure communication between heterogeneous cryptosystems.

#### Performance analysis

Our scheme is compared with the schemes in references [25, 27, 28] in terms of performance. Reference [28] uses the traditional public key encryption scheme. We use a PC equipped with Intel Core i7–7500u CPU@3.5GHz, 8G memory, and Windows 10 for simulation. The representative symbols and their meaning and computational time are shown in Table 4. The computational cost of each comparison scheme is shown in Table 5. With the increase of plaintexts/ctipertexts, the computational costs of our scheme and the comparison schemes in

---

Table 3. Characteristics of our scheme.

| Scheme   | ET | MET | SC | HSC | Without bilinear pairing | Cryptosystem | Confidentiality | Unforgeability |
|----------|----|-----|----|-----|--------------------------|--------------|----------------|----------------|
| Our scheme | ✓  | ✓   | ✓  | ✓   | ✓                        | PKI—CLC      | IND-CCA2       | EUF-CMA        |

✓: not supported;  
✓: supported.

https://doi.org/10.1371/journal.pone.0274695.t003
In the signcryption/encryption phase, it can be seen from Table 5 and Fig 2 that compared with the schemes in [25] and [27], our scheme does not have bilinear pairing operations, which greatly reduces the computational cost. Although compared with the scheme in [28], the computational cost of our scheme is higher, our scheme not only achieves confidentiality, but also satisfies non-repudiation. And our scheme supports communication between heterogeneous cryptosystem. In the unsigncryption/decryptio n and test phases, Figs 3 and 4 clearly show that our scheme has lower computational costs than the schemes in [25, 27, 28]. When the number of ciphertexts reaches 20, the computational efficiency in unsigncryption/decryption phase of our scheme is approximately 2000 times, 1500 times and 30 times that of the three comparison schemes. And as the number of ciphertexts increases, the advantages of our scheme become more obvious in the test phase.

### Table 4. Notations.

| Symbols | Representations | Time( ms) |
|---------|-----------------|-----------|
| \( k \) | The number of ciphertexts that can run the Test algorithm | | |
| \( T_h \) | The time it takes to run a hash operation | 0.0014 |
| \( T_a \) | The time it takes to run a point-add operation in group \( G_1 \) | 1.3667 |
| \( T_m \) | The time it takes to run a multiplicative operation in group \( G_1 \) | 0.0032 |
| \( T_e \) | The time it takes to run an exponential operation in group \( G_2 \) | 0.2549 |
| \( T_p \) | The time it takes to run a bilinear pairing operation | 6.9841 |

### Table 5. Comparison of computational cost.

| Schemes | Signcryption/Encryption | Unsigncryption/Decryption | Test |
|---------|--------------------------|---------------------------|------|
| [25]    | \( 5T_h + 5T_a + 2T_e = 7.3503\)ms | \( 4T_h + T_e + 3T_p = 21.2122\)ms | \( 2T_h + 2T_e + 2T_p = 14.4808\)ms |
| [27]    | \( 4T_h + T_a + 2T_e + 5T_m + 3T_p = 15.8663\)ms | \( 4T_h + 5T_e + 2T_p = 15.2483\)ms | \( 2T_h + 2T_m + 4T_p = 27.9456\)ms |
| [28]    | \( 4T_h + 3T_e = 0.0073\)ms | \( 3T_h + 2T_e = 0.514\)ms | \( 2T_h + 2T_e = 0.5216\)ms |
| ours(k = 1) | \( 5T_h + T_a + 5T_m = 1.3896\)ms | \( 5T_h + 3T_m = 0.0166\)ms | \( 2T_h + 4T_m = 0.0156\)ms |

The signcryption/encryption, unsigncryption/decryption, and test phases are shown in Figs 2–4 respectively.

In the signcryption/encryption phase, it can be seen from Table 5 and Fig 2 that compared with the schemes in [25] and [27], our scheme does not have bilinear pairing operations, which greatly reduces the computational cost. Although compared with the scheme in [28], the computational cost of our scheme is higher, our scheme not only achieves confidentiality, but also satisfies non-repudiation. And our scheme supports communication between heterogeneous cryptosystem. In the unsigncryption/decryption and test phases, Figs 3 and 4 clearly show that our scheme has lower computational costs than the schemes in [25, 27, 28]. When the number of ciphertexts reaches 20, the computational efficiency in unsigncryption/decryption phase of our scheme is approximately 2000 times, 1500 times and 30 times that of the three comparison schemes. And as the number of ciphertexts increases, the advantages of our scheme become more obvious in the test phase.
Conclusion

We proposed the HSC-MET scheme to overcome the problems in the existing schemes, such as not supporting the communication between heterogeneous cryptosystems, high computational overhead, and low efficiency of ciphertext retrieval. Our scheme uses HSC technology to realize secure communication from PKI to CLC. The scheme has no bilinear pairing operation, which greatly reduces the computational cost and improves communication efficiency. In addition, the multi-ciphertext equality test technology is introduced to realize the simultaneous retrieval of multiple ciphertexts by multiple users, which reduces the computational cost of the ciphertext equality test in the multi-user scenario. Under the ROM, we proved the confidentiality, unforgeability, and number security of the HSC-MET scheme based on the CDH problem. Finally, we compared the scheme with several similar schemes. The results show that our scheme not only has more functional features and higher security but also has lower computational costs in signcryption, unsigncryption, and test phases. However, our scheme’s security is proved under the random oracle model, which is not universal in reality more or less. In the future, we will further investigate the security under the standard model to make the HSC-MET scheme more practical.
Supporting information

S1 Fig. Cloud server. Image URL: https://www.iconfont.cn/search/index?searchType=iconq=cloud.
(TIF)

S2 Fig. KGC. Image URL: https://www.iconfont.cn/search/index?searchType=iconq=server.
(TIF)

S3 Fig. UAVs. Image URL: https://www.iconfont.cn/search/index?searchType=iconq=UAV.
(TIF)

S4 Fig. Data users. Image URL: https://www.iconfont.cn/search/index?searchType=iconq=user.
(TIF)

S5 Fig. CA. Image URL: https://www.iconfont.cn/search/index?searchType=iconq=host.
(TIF)

Author Contributions

Writing – original draft: Xiaodong Yang, Ningning Ren.

Writing – review & editing: Xiaodong Yang, Ningning Ren, Aijia Chen, Zhisong Wang, Cai-fen Wang.

References

1. Liu Y, Dai HN, Wang Q, Shukla MK, Imran M. Unmanned aerial vehicle for internet of everything: opportunities and challenges. Computer Communications. 2020; 155:66–83. https://doi.org/10.1016/j.comcom.2020.03.017

2. Wang BH, Wang DB, Ali ZA, Ting BT, Wang H. An overview of various kinds of wind effects on unmanned aerial vehicle. Measurement and Control. 2019; 52(7-8): 731–739. https://doi.org/10.1177/0020294019847688

3. Boccadore P, Striccoli D, Grieco LA. An extensive survey on the internet of drones. Ad Hoc Networks. 2021; 122:102600. https://doi.org/10.1016/j.adhoc.2021.102600

4. Gharibi M, Boutaba R, Waslander SL. Internet of drones. IEEE Access. 2016; 4: 1148–1162. https://doi.org/10.1109/ACCESS.2016.2537208

5. Yahuza M, Idris M, Ahmedy IB, Wahab A, Bala A. Internet of drones security and privacy issues: taxonomy and open challenges. IEEE Access. 2021; 9:57243–57270. https://doi.org/10.1109/ACCESS.2021.3072030

6. Srinivas J, Das AK, Kumar N, Rodrigues JJPC. TCALAS: temporal credential-based anonymous lightweight authentication scheme for internet of drones environment. IEEE Transactions on Vehicular Technology. 2019; 68(7):6903–6916. https://doi.org/10.1109/TVT.2019.2911672

7. Bharany S, Sharma S, Bhatia S, MKI Rahmani, Shuaib M, et al. Energy Efficient Clustering Protocol for FANETS Using Moth Flame Optimization. Sustainability 2022; 14(10):6159. https://doi.org/10.3390/su14106159

8. Bharany S, Sharma S, Frinda J, Shuaib M, Khalid MI, et al. Wildfire Monitoring Based on Energy Efficient Clustering Approach for FANETS. Drones. 2022; 6(8):193. https://doi.org/10.3390/drones6080193

9. Bera B, Chattaraj D, Das AK. Designing secure blockchain-based access control scheme in IoT-enabled internet of drones deployment. Computer Communications. 2020; 153:229–249. https://doi.org/10.1016/j.comcom.2020.02.011

10. Hussain S, Chaudhry SA, Alomari OA, Alsharif MH, Khan MK, Kumar N. Amassing the security: an ECC-based authentication scheme for internet of drones. IEEE Systems Journal. 2021; 15(3):4431–4438. https://doi.org/10.1109/JSYST.2021.3097047
11. Khan M, Shah H, Rehman S, Kumar N, Ghazali R, Shehzad D, et al. Securing internet of drones with identity-based proxy signcryption. IEEE Access. 2021;p: 89133–89142. https://doi.org/10.1109/ACCESS.2021.3089009

12. Gope P, Sikdar B. An efficient privacy-preserving authenticated key agreement scheme for edge-assisted internet of drones. IEEE Transactions on Vehicular Technology. 2020; 69(11):13621–13630. https://doi.org/10.1109/TVT.2020.3018778

13. Sun Y, Li H. Efficient signcryption between TPKC and IDPKC and its multi-receiver construction. Science China Information Sciences. Science China Information Sciences. 2010; 53(3):557–566. https://doi.org/10.1007/s11432-010-0061-5

14. Elkhaili A, Zhang J, Elhabob R, Eltayieb N. An efficient signcryption of heterogeneous systems for Internet of Vehicles. Journal of Systems Architecture. 2021; 113:101885. https://doi.org/10.1016/j.sysarc.2020.101885

15. Ali I, Lawrence T, Omala AA, Li F. An efficient hybrid signcryption scheme with conditional privacy-preservation for heterogeneous vehicular communication in VANETs. IEEE Transactions on Vehicular Technology. 2020; 69(10):11266–11280. https://doi.org/10.1109/TVT.2020.3008781

16. Qiu J, Fan K, Zhang K, Pan Q, Yang Y. An efficient multi-message and multi-receiver signcryption scheme for heterogeneous smart mobile IoT. IEEE Access. 2019; 7:180205–180217. https://doi.org/10.1109/ACCESS.2019.2958089

17. Cao S, Lang X, Liu X, Zhang Y, Wang C. Improvement of a provably secure mutual and anonymous heterogeneous signcryption scheme between PKI and IBC. Journal of Electronics Information Technology. 2019; 41(8):1787–1792.

18. Li W, Jin C, Kumari S, Xiong H, Kumar S. Proxy re-encryption with equality test for secure data sharing in Internet of Things-based healthcare systems. Transactions on Emerging Telecommunications Technologies. 2020; 2:e3986.

19. Xiong H, Zhao Y, Hou Y, Huang X, Jin C, Wang L, et al. Heterogeneous signcryption scheme supporting multi-ciphertext equality test for Internet of Drones. IEEE Internet of Things Journal. 2021; 8(21):16142–16152. https://doi.org/10.1109/JIOT.2020.3008955

20. Chen J, Wu G, Su X, Zhang Y. An efficient certificateless public key cryptography with equality test for Internet of Vehicles. IEEE Access. 2019; 7:68957–68969. https://doi.org/10.1109/ACCESS.2019.2917326

21. Xiong H, Hou Y, Huang X, Zhao Y, Chen CM. Heterogeneous signcryption scheme with equality test from IBC to PKI. IEEE Systems Journal. 2021;(99):1–10.

22. Hou Y, Huang X, Chen Y, Kumar S, Xiong H. Heterogeneous signcryption scheme supporting equality test from PKI to CLC toward IoT. Transactions on Emerging Telecommunications Technologies. 2021; 32(8):e4190. https://doi.org/10.1002/ett.4190

23. Choi S, Lee HT. Attack and improvement of the recent identity-based encryption with authorized equivalence test in cluster computing. Cluster Computing 2022; 25(1):633–646. https://doi.org/10.1007/s10586-021-03409-x