Research Article

An Improved Efficient Certificateless Hybrid Signcryption Scheme for Internet of Things

Yong Wu,1 Bei Gong,1 and Yu Zhang1,2

1Beijing Key Laboratory of Trusted Computing, Faculty of Information Technology College of Computer Science and Technology, Beijing University of Technology, 100124 Beijing, China
2School of Information Science and Technology, Zhengzhou Normal University, Henan 450044, China

Correspondence should be addressed to Bei Gong; gongbei@bjut.edu.cn

Received 20 December 2021; Accepted 24 March 2022; Published 19 April 2022

Academic Editor: Yushu Zhang

Copyright © 2022 Yong Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet of Things (IoT) is a vast perceptual network formed by a variety of IoT devices connected to each other. In IoT, various devices cooperate with each other to collect and transmit private data of IoT and sustain the efficient and orderly operation of IoT. However, in the complex security situation of IoT, due to the resources of IoT devices being limited, it is difficult for IoT devices to afford the heavy resource consumption of sophisticated data encryption and decryption mechanism, which brings certain security risk to data transmission of IoT. To ensure safe and efficient data transmission, we propose an improved efficient certificateless hybrid signcryption scheme for IoT, which satisfies confidentiality, unforgeability, forward security, public verification, and known session-specific temporary information security; meanwhile, we prove them in random oracle model (ROM). In addition, through simulation experiment, we find that our scheme has higher communication efficiency and computational efficiency compared with existing schemes.

1. Introduction

The Internet of Things (IoT) realizes the mapping from the real world to the virtual world through the perception of a large number of IoT devices to the real world. In the virtual world, by sorting out and analyzing massive perceptual data, users can obtain more accurate and effective information, so as to obtain detailed status in the real world, and then realize more accurate and efficient adjustment to the real world. At present, with the extensive development of the IoT, IoT has brought unimaginable convenience to the people and the country. For example, in the smart home, various furniture and equipment can cooperate with each other, so that people can get rid of heavy housework [1]. In the smart medical, through the real-time perception and analysis of the disease, patients can attain more perfect and comprehensive treatment in time [2]. Therefore, IoT plays a key role in people’s production and life and has become one of the most potential technologies in the 21st century [3].

In IoT, there are a large number of devices with different structures, which cooperate to transmit sensitive data or private information of IoT, forming an organic whole. However, in order to reduce production costs, manufacturers often adopt overly simple structure design and limited resources in the design and manufacture of IoT devices. It is easy for attackers to guess the general structure of the device and carry out illegal operations such as hacking and misappropriation [4]. Additionally, in the face of the complex security situation of IoT, it is difficult for the resource-constrained devices to afford the heavy resource consumption of sophisticated data transmission mechanism, which also brings certain security risks to the data transmission in IoT [5]. Therefore, for the massive resource-constrained devices in IoT, how to ensure the efficient and secure data transmission has become a key factor in the security of IoT [6].

Data encryption has become the primary choice to ensure the security of data transmission and has been suc-
cessfully applied in many fields such as the Internet. However, in IoT, the normal data encryption mechanism is no longer suitable for data encryption of IoT devices because it consumes too much software and hardware resources or lacks some security features of IoT. In order to reduce the resource consumption in the process of data encryption and improve efficiency and security levels of data transmission, the researchers put forward a series of lightweight data encryption algorithm [7, 8]. Among them, the concept of certificateless hybrid signcryption has attracted extensive attention since it was put forward.

Compared with other data encryption methods, the certificateless hybrid signcryption mechanism can greatly reduce the consumption of computing resources and communication resources in the process of data transmission and has higher flexibility and security [9]. In the certificateless cryptosystem, the key generation center (KGC) can generate the user’s private key according to the user’s public information and send it to the user through the secure channel. After receiving the partial private key, the user combines its own secret value to generate the full private key. After that, the user computes the public key to complete the initialization of the user key. Compared with other public key cryptosystem, the certificateless cryptosystem gets rid of the problem of public key authentication in traditional cryptosystem based on public key infrastructure (PKI). Meanwhile, the full private key of the users in the certificateless public key cryptography is generated jointly by the user and KGC and overcomes the key escrow problem in identity-based cryptography [10]. In terms of data encryption algorithm, compared with the traditional “signature before encryption” method, the hybrid signcryption algorithm can achieve the signature and encryption in one calculation turn, which can improve the computing efficiency and provide stronger security for data transmission [11]. Therefore, for resource-constrained devices and complex security situations in IoT, the certificateless hybrid signcryption mechanism can not only greatly reduce the consumption of computing resources and the communication pressure in data transmission but also can provide better security and flexibility for data transmission of IoT [12].

However, in the practical application of resource-constrained devices of IoT, the existing certificateless hybrid signcryption schemes are hard to keep the balance between security and efficiency. In security, the capabilities of attackers are also increasing day by day in IoT, and some proposed schemes have certain flaws in the face of new security requirements of IoT [6]. In efficiency, bilinear pairing operation improves the flexibility and security of the algorithm but also increases the computational pressure on devices [13]. However, most existing schemes adopt multiple bilinear pairing operations, which consume too much hardware and software resources and are no longer suitable for the data transmission requirements of resource-constrained devices in IoT. Therefore, to ensure secure and efficient data transmission in IoT, our contribution is therefore as follows.

(1) We evaluate the security of Gong et al.’s certificateless hybrid signcryption scheme for IoT [14]

(2) We propose an improved efficient certificateless hybrid signcryption scheme for IoT

(3) Our scheme can meet confidentiality, unforgeability, forward security, public verification, and known session-specific temporary information security, and we prove them in random oracle model (ROM)

(4) In theoretical analysis and experimental simulation, we find that our scheme has higher communication efficiency and computational efficiency than the existing schemes

The rest of the paper is organized as follows. In Section 2 and Section 3, we present research status and preliminaries related to this paper. The review and the evaluation of the Gong’s scheme are carried out in Section 4. In Section 5, we show the specific details of our scheme. In Section 6, we analyze the security of the proposed scheme. In Section 7, we perform theoretical analysis and experimental simulation. In Section 8, we make the summary of this paper.

2. Related Work

Since the concept of certificateless hybrid signcryption is put forward, it has attracted more and more attention, and a series of effective schemes had been produced. Li et al. took the lead in proposing the concept of certificateless hybrid signcryption, gave an example of certificateless hybrid signcryption scheme, and declared that their scheme satisfied confidentiality and unforgeability in ROM [9]. However, Li et al.’s scheme uses too many bilinear pairs, resulting in low computational efficiency.

To ensure efficient data transmission, a series of feasible methods and schemes were proposed. Yin and Liang proposed a certificateless hybrid signcryption scheme for wireless sensor networks, which performed only two bilinear calculations and improved computational efficiency [15]. However, after receiving valid message, the receiver can forge any information, resulting in Yin and Liang’s scheme not meeting the unforgeability. Yu and Yang proposed a certificateless hybrid signcryption scheme without pairing, which did not require bilinear computation and improved computing efficiency [16]. However, compared with other schemes, this scheme transmits more data and has certain deficiencies in communication efficiency. Luo and Ma proposed a new certificateless hybrid signcryption scheme for cloud storage, which did not require bilinear pairings and reduced the computational pressure in the process of data encryption [17]. However, Kasyoka et al. proved that Luo and Ma’s scheme did not meet the requirement of unforgeability and had certain security defects and proposed an improvement scheme [18].

With the gradual improvement of the attacker’s ability, researchers also put forward some certificateless hybrid signcryption schemes to enhance the security of data transmission. Luo et al. proposed a certificateless hybrid signcryption scheme, which can meet confidentiality, unforgeability, and the known session-specific temporary information security [19]. However, this scheme uses too many
bilinear operations and has certain deficiencies in computing efficiency. To ensure data security and low computational overhead, Gong et al. proposed a lightweight and secure certificateless hybrid signcryption for the IoT, but it cannot satisfy the requirements of unforgeability, and its security analysis will be explained in Section 4 of this paper [14].

3. Preliminaries

3.1. Formal Model of Certificateless Hybrid Signcryption. There are three members: KGC, sender, and receiver. The general interaction process is as follows.

Setup. The KGC randomly chooses master system key $s$ and generates system parameters $\params$. After the algorithm is executed, KGC publishes the system parameters $\params$ to all users

Extract Partial Private Key. The user firstly sends his unique identifier $ID_i$ to KGC, which generates the partial private key $d_i$ of user and sends the partial private key $d_i$ to the user through secure channel

Generate User Key. After receiving the system parameters published by KGC, the user generates the secret value $x_i$ and generates the public key $P_i$. Next, the user publishes the public key $P_i$ and reserves the secret value $x_i$

Extract Full Private Key. After receiving partial private key $d_i$ sent secretly by KGC, the user generates the full private key $s_i$ combined with the secret value $x_i$. From there, the user and KGC jointly generate the full private keys $s_i$ of user

Signcryption. Assume that both sender and receiver have completed the initialization of the key. During the signcryption process, the sender generates the session key $K$, encrypts the information $m$ with symmetric encryption algorithm, and generates ciphertext $\sigma$. After the signcryption is complete, the sender sends the ciphertext $\sigma$ to the receiver

Unsigncryption. After receiving the ciphertext $\sigma$, the receiver generates the session key $K$ and decrypts the ciphertext to recover information $m$ or error symbols $\bot$. Finally, the receiver executes the authentication equation to judge the validity of the ciphertext

3.2. Security Notions. The security of modern cryptographic systems mostly is based on mathematical problems, and this paper is no exception. The mathematical difficult problems proposed in this paper are involved as follows.

Definition 1. Let us say that $G_1$ is an additive cyclic group and $P$ is the generator of $G_1$. The challenger chooses $aP, bP, T \in G_1$, where $a, b, c \in \mathbb{Z}_q^*$. The computational Diffie-Hellman (CDH) is to compute $abP$.

Definition 2. Let us say that $G_2$ is a multiplication cyclic group and $P$ is the generator of $G_2$. The challenger chooses $P^a, P^b$, where $a, b \in \mathbb{Z}_q^*$. The computational Diffie-Hellman (CDH) is to compute $P^{ab}$.

Definition 3. Let us say that $G_1$ is an additive cyclic group and $P$ is the generator of $G_1$. The challenger chooses $aP, bP, T \in G_1$, where $a, b, c \in \mathbb{Z}_q^*$. The decisional Diffie-Hellman (DDH) is to determine whether the equation $abP = T$ is true.

Definition 4. Let us say that $G_1$ is an additive cyclic group, $G_2$ is a multiplication cyclic group, $\tilde{e} : G_1 \times G_1 \rightarrow G_2$ is a bilinear pairing, and $P$ is the generator of $G_1$. The challenger chooses $aP, bP, cP \in G_1$ and $T \in G_2$, where $a, b, c \in \mathbb{Z}_q^*$. The decisional bilinear Diffie-Hellman (DBDH) is to determine whether the equation $\tilde{e}(P, P)^{ab} = T$ is true.

4. Gong et al.’s Scheme Review and Security Analysis

In this section, we review the scheme proposed by Gong et al. and show the security analysis of Gong et al.’s scheme [14]. In the security analysis, we find that the scheme does not meet the unforgeability and prove it.

4.1. The Review of Gong et al.’s Scheme. Assuming that Alice is the sender and Bob is the receiver in the communication process, then the details of Gong et al.’s scheme are as follows.

Setup. After receiving the system parameter $y$, the KGC selects an additive cyclic group $G_1$, a multiplicative cyclic group $G_2$, a bilinear pairing $\tilde{e} : G_1 \times G_1 \rightarrow G_2$, and $P$ as the generator of the additive cyclic group $G_1$, where $|G_1| = |G_2| = q$. Then, KGC selects four hash functions $h_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$, $h_2 : \{0, 1\}^* \times G_1 \rightarrow \mathbb{Z}_q^*$, $h_3 : G_2 \times \mathbb{Z}_q^* \rightarrow \{0, 1\}^n$, and $h_5 : G_1 \rightarrow \mathbb{Z}_q^*$ and one modular function $h_4 : amodb, b = h_4(R)$. After that, KGC selects $x \in \mathbb{Z}_q^*$ as the master system key and computes $P_{pub} = xP$ as the system public key. Finally, KGC publishes system parameters and keeps the master system key $x$ secret

Extract Partial Private Key. After receiving the user’s unique identifier $ID_i \in \{0, 1\}^r$ sent by the user $u_i$, and KGC computes $d_i = xQ_i$, as the partial private key of the user $u_i$, where $Q_i = h_1(ID_i)$. Next, KGC sends the partial private key $d_i$ to the user $u_i$ through the private channel

Generate User Key. The user selects a random value $x_i$ as the secret value and computes $P_i = x_iP$ as the public key. After receiving the partial private key $d_i$ of the user sent by KGC, the user generates the full private key $s_i = (x_i, d_i)$

Signcryption. Suppose Alice and Bob have completed the initialization of the key, and Alice already gets Bob’s public key $P_B$. The specific process of signcryption is as follows.

(1) Chooses $r \in \mathbb{Z}_q^*$ randomly and compute $R = rP$

(2) Computes $y = P_B^{x_ih_5(R)}$ and $z = h_3(Q_Bd_A)$, where $Q_B = h_1(u_B)$

(3) Computes the session key $K = h_4(y, z, R)$

(4) Generates ciphertext $c = Enc_{K}(m)$
(5) Computes \( f = h_2(u_A, u_B, R, P_A, P_B) \) and \( s = rz/s_Af \) and sends the ciphertext \( \sigma = (c, R, s) \) to Bob.

Unsigncryption. Assume that Bob has obtained the public key \( P_A \) of the sender Alice and received the signcryption ciphertext \( \sigma = (c, R, s) \). The specific process of unsigncryption is as follows:

1. Computes \( y = P_A^y h_1(R) \) and \( z = h_3(Q_A d_B) \), where \( Q_A = h_1(u_A) \).
2. Computes the session key \( K = h_4(y, z, R) \) and generates the message \( m \setminus \perp = \text{Dec}_K(c) \).
3. Computes \( f = h_2(u_A, u_B, R, P_A, P_B) \) and checks if the verification equation \( P_A^{sf} = \bar{e}(zP, R) \) is true.
4. If true, Bob receives the message. If not, Bob drops the message.

4.2. Security Analysis. Through the analysis of the Gong et al.'s scheme, we find that it cannot meet the unforgeability. It is assumed that the sender Alice and the receiver Bob have generated their key. If the receiver Bob has received a valid ciphertext \( \sigma = (c, R, s) \), then Bob is able to forge any ciphertext \( \sigma' \) and claims that it was sent by the sender Alice. The forgery process is as follows:

1. Chooses \( a \in Z_q^* \) and \( R' = Ra = arP \).
2. Computes \( y' = P_A^y h_1(R') \) and \( z = h_3(Q_A d_B) \).
3. Computes the session key \( K = h_4(y', z, R') \) and executes the symmetric encryption algorithm to generate the ciphertext \( c = \text{Enc}_K(m') \), where \( m' \) is the information Bob wants to get.
4. Computesthe \( f = h_2(u_A, u_B, R, P_A, P_B) \) from the previous communication and the forgeable \( f' = h_2(u_A, u_B, R', P_A, P_B) \).
5. Computes \( s' = asf^{f'-1} \) and generate the valid forgeable ciphertext \( \sigma' = (c', R', s') \).

The forgeable ciphertext \( \sigma' = (c', R', s') \) can be able to overcome the verification equation. The verification process is as follows:

\[
P_A^{sf'} = P_A^{asf^{f'-1}f} = P_A^{asf} = \bar{e}(P, P)^{sA} = \bar{e}(zP, P)^{ar} = \bar{e}(zP, R').
\]

Since then, the receiver Bob has successfully forged a valid signcryption ciphertext. Thus, Gong et al.'s proposed scheme does not satisfy the requirement of unforgeability. Based on the analysis of the Gong et al.'s scheme, we propose an improved scheme, which will be introduced in Section 5.

### Table 1: The symbol description table.

| Symbol  | Description      |
|---------|------------------|
| \( p \) | The generator of \( G_i \) |
| \( s \) | Master system key |
| \( P_{pub} \) | System public key |
| \( SK_i = (x_i, D_i) \) | User’s full private key |
| \( PK_i = (P_i, R_i) \) | User’s full public key |
| \( h_1, h_2, h_3 \) | Hash function |

5. The Proposed Scheme

In this section, we take the communication between Alice and Bob as an example to introduce our scheme in detail, where Alice represents the sender and Bob represents the receiver. For increasing readability, the main symbols involved in our scheme are introduced in Table 1.

**Setup.** Let \( G_1 \) and \( G_2 \) be an additive cyclic group and a multiplicative cyclic group, where \( |G_1| = |G_2| = p \) and \( p \) is a prime. Choose a bilinear pairing \( \bar{e} : G_1 \times G_1 \rightarrow G_2 \), \( P \) be the generator of \( G_1 \) and a pair of symmetric encryption and decryption algorithms (Enc, Dec). Select three hash functions \( h_1, h_2, h_3 : \{0, 1\}^l \rightarrow Z_q^* \), \( h_2 = G_1 \times G_2 \rightarrow G_1 \rightarrow \{0, 1\}^l \), and \( h_3 = \{0, 1\}^l \times \{0, 1\}^l \times G_2^2 \times G_1^4 \rightarrow Z_q^* \), where \( l \) is the length of identity and \( n \) is the length of session key, respectively. Then, the KGC chooses \( s \in Z_q^* \) as the master system key and computes \( P_{pub} = sP \) as the system public key. Finally, the KGC publishes the system parameters \( \text{params} = \{G_1, G_2, q, \bar{e}, P, P_{pub}, h_1, h_2, h_3, \text{Enc, Dec}\} \) and keeps the master system key \( s \) secret.

**Extract Partial Private Key.** After receiving the unique identifier \( ID_i \in \{0, 1\}^4 \) of user, the KGC randomly selects \( r_i \in Z_q^* \), computes \( Q_i = h_1(ID_i), R_i = r_iP \), and generates the partial private key \( D_i = r_i + s \cdot h_1(ID_i) \). Finally, KGC sends the partial private key \( D_i \) and \( R_i \) to the user.

**Generate User Key.** After receiving the partial private key \( D_i \) and the partial public key \( R_i \), the user randomly chooses \( x_i \in Z_q^* \) as the secret value and computes \( P_i = \bar{e}(P, P_{pub})^{x_i} \) as the public key. Note that the user combines \( R_i \) and \( P_i \) to generate the full public keys \( PK_i = (R_i, P_i) \).

**Extract the Full Private Key.** The user combines the secret value \( x_i \) and the partial private key \( D_i \) to generate the full private key \( SK_i = (x_i, D_i) \).

**Signcryption.** Before performing signcryption, the sender Alice has obtained the full public key \( PK_A = (R_A, P_A) \), the full private key \( SK_A = (x_A, D_A) \), the message \( m \) needed to be transmitted, system parameters params, the full public key \( PK_B = (R_B, P_B) \), and identity \( ID_B \) of the receiver Bob. The detailed process is as follows:

1. Chooses \( r \in Z_q^* \) randomly and compute \( R = rP \).
2. Computes \( t_1 = (R_B + P_{pub}Q_B)r \) and \( t_2 = P_B^{x_B} \), where \( Q_B = h_1(ID_B) \).
Table 2: Comparison of message lengths during communication.

| Scheme                        | Ciphertext length |
|-------------------------------|--------------------|
| Luo et al.’s scheme [19]      | $q + 2g_1 + m = 350$ bits |
| Yin and Liang’s scheme [20]   | $2q + g_1 + m = 305$ bits |
| Our scheme                    | $q + g_1 + m = 285$ bits |

(3) Computes session key $k = h_3(t_1, t_2, R)$ and generates the ciphertext $c = Enc_{k}(m)$

(4) Computes $h = h_5(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R)$ and $s = (r + hD_A)/x_A$

(5) Sends ciphertext $\sigma = (c, R, s)$ to receiver Bob

Unsignification. Before performing unsignification, the receiver Bob has obtained the full public key $PK_B = (R_B, P_B)$, the full private key $SK_B = (x_B, D_B)$, the ciphertext $\sigma = (c, R, s)$, the system parameters $\sigma$, the public key $P$ $K_A = (R_A, P_A)$, and identity $ID_A$ of the sender Alice. The detailed process is as follows.

(1) Computes $t_1 = RD_B$ and $t_2 = P_A^{x_A}$

(2) Computes session key $k = h_2(t_1, t_2, R)$ and generate message $m = Dec_{k}(c)$ or error symbol ⊥

(3) Computes $h = h_5(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R)$

(4) Checks if the equation $P_{pub} = \bar{\varepsilon}(R + hR_A + hQ_AP_{pub}, P_{pub})$ holds. If true, the Bob receives the message $m$; otherwise, the message is dropped.

5.1. Correctness.

$$
t_1 = (R_B + P_{pub}Q_B)r = (r_BP + sQ_B)P = R_BR,$$

$$
t_2 = P_A^{x_A} = \bar{\varepsilon}(P_{pub})P_{pub}^{x_A} = \bar{\varepsilon}(P_{pub})P_{pub}^{x_A} = P_A^{x_A},
$$

$$
P_A^{x_A} = P_A^{(r + hD_A)/x_A} = \bar{\varepsilon}(P_{pub})P_{pub}^{x_A(r + hD_A)/x_A} = \bar{\varepsilon}(P_{pub})^{r + hD_A}
$$

$$
\bar{\varepsilon}(R + rD_A)P_{pub} = \bar{\varepsilon}(R + hD_A)P_{pub} = \bar{\varepsilon}(R + hA + sh_1(ID_A))P{pub} = \bar{\varepsilon}(R + hR_A + hQ_AP_{pub}, P_{pub}).
$$

6. Security Analysis

There are two kinds of security adversary $A_1, A_2$ in certificateless public key cryptosystems. The adversary $A_1$, impersonating the user in cryptosystems, can replace the public key of any user in system. The adversary $A_2$, impersonating the KGC in cryptosystems, can obtain the master system key. In this section, we discuss the security attributes of our scheme from the perspective of two types of attackers.

6.1. Confidentiality

**Lemma 5.** Assuming that the adversary $A_1$ can be able to win the game $\text{Game}_{A_1}^{\text{IND}-\text{CCA}^1}$ with a nonnegligible advantages $\epsilon$ after $q_{h_1}$ queries, $q_{h_2}$ queries, $q_{h_3}$ queries, $q_{pk}$ public key queries, $q_{dk}$ secret value queries, $q_{pk}$ the partial private key queries, and $q_{pk}$ the replace public key queries, the challenger $C$ can solve the CDH problem with the follow probability $\epsilon'$. Note that $q_{T} = q_{1} + q_{dk} + q_{pk} + q_{pk}$.

$$
\epsilon' = \epsilon \left( 1 - \frac{q_{pk}}{q_{T}} \right) \left( 1 - \frac{q_{dk}}{q_{T}} \right) \left( 1 - \frac{q_{pk}}{q_{T}} \right) \left( 1 - \frac{q_{pk}}{q_{T}} \right) \left( 1 - \frac{q_{pk}}{q_{T}} \right). \quad (3)
$$

**Proof.** The challenger $C$ takes adversary $A_1$ as the subroutine, gives the adversary $A_1$ the $(P, aP, bP)$. And the target is compute the $abP$ through the game between adversary $A_1$ and the challenger $C$.

Setup. The challenger $C$ executes the setup algorithm and generates the system public parameters $\sigma$.

Phase 1. The adversary $A_1$ execute the following polynomially bounded queries. Note that before the adversary executes queries, the challenger $C$ chooses randomly $I, J \in [1, q_1]$ for the challenged phase.

- $h_1$ query: the challenger $C$ initializes and updates data table $L_1$, where the tuple format is $(ID_i, I)$. When adversary $A_1$ executes the $h_1$ query with the user identity $ID_i$ as input, $C$ firstly checks if $L_1$ contains item $(ID_i, I)$. If $L_1$ contains the tuple $(ID_i, I)$, $C$ returns the $I$ to $A_1$. If not, $C$ randomly chooses the $I \in \mathbb{Z}_q^*$, adds or updates the new item $(ID_i, I)$ to $L_1$, and returns $I$ to $A_1$.

- $h_2$ query: the challenger $C$ initializes and updates data table $L_2$, where the tuple format is $(t_1, t_2, R, k)$. When adversary $A_1$ executes the $h_2$ query with the $(t_1, t_2, R)$ as the input, the challenger $C$ firstly checks if $L_2$ contains the related tuple $(t_1, t_2, R, k)$. If $L_2$ contains the tuple $(t_1, t_2, R, k)$, $C$ returns the $k$ to $A_1$. If not, $C$ randomly chooses $k \in \{0, 1\}^*$, inserts the new item $(t_1, t_2, R, k)$ to the list $L_2$, and returns the $k$ to $A_1$.

- $h_3$ query: the challenger $C$ initializes and updates data table $L_3$, where the tuple format is $(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R, h)$. When the adversary $A_1$ executes the $h_3$ query with the $(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R)$ as input, the challenger $C$ firstly checks if $L_3$ contains the interrelated tuple. If $L_3$ contains this tuple, $C$ returns the $h$ to $A_1$. If not, $C$ randomly chooses $h \in \mathbb{Z}_q^*$, adds or updates the new item $(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R, h)$ to $L_3$, and returns the $h$ to $A_1$.

Secret value query: the challenger $C$ initializes and updates data table $L_k$, where the tuple format is $(ID_i, P_i, R_i, x_i)$. When adversary $A_1$ executes the secret value query with the $ID_i$ as the input, the challenger $C$ firstly checks if $L_k$ contains the interrelated tuple $(ID_i, P_i, x_i)$. If $L_k$ contains this tuple, $C$ returns the $x_i$ to $A_1$. If not, $C$ chooses $y_i \in \mathbb{Z}_q^*$ randomly, computes $P_i = \bar{\varepsilon}(P_{pub})^y_i$, adds or updates the item $(ID_i, P_i, x_i)$ to $L_k$, and returns $x_i$ to $A_1$. Note that if $ID_i \neq ID_j$, $C$ aborts this game.

Partial private key query: with the $ID_i$ as the input, the challenger $C$ firstly checks if $L_k$ contains the interrelated tuple $(ID_i, P_i, x_i)$. If $L_k$ contains this tuple, $C$ returns the $D_1$ to $A_1$. Otherwise, if $ID_i \neq ID_j$, $C$ chooses $r_i \in \mathbb{Z}_q^*$ and
computes $R_i = r_i P + D_i = r_i + s_i$, where $l_i$ is from the list $L_i$. Next, $C$ adds or updates the item $(ID_A, R_i, r_i, s_i)$ to $L_k$ and returns $D_i$ to $A_i$. Note that if $ID_A = ID_B$, $C$ aborts this game.

Public key query: when adversary $A_i$ executes the public key query with the $ID_A$ as the input, the challenger $C$ firstly checks if $L_k$ contains list item $(ID_A, P_i, R_i, r_i, s_i)$. If $L_k$ contains this tuple, $C$ returns the $PK_i = (R_i, P_i)$ to $A_i$. Otherwise, if $ID_A \neq ID_B$, $C$ executes the secret value query and partial private key query. If $ID_A = ID_B$, $C$ sets $x_i \in Z_q^*$, $D_i = b \in G_1$ and computes $P_i = e(R_i, P_{pub})^{y_i}$ and $R_i = b - 1, P_{pub}$, where $l_i$ is from the list $L_i$. Next, $C$ adds or updates the item $(ID_A, P_i, R_i, r_i, s_i)$ to $L_k$ and returns $PK_i = (R_i, P_i)$ to $A_i$.

Replace public key query: with the $ID_A, PK_i = (R_i, P_i)$ as the input, $C$ updates the list $L_k$ with the new item $(ID_A, R_i, P_i, R_i, r_i, s_i)$.

Signcryption query: when adversary $A_i$ executes the signcryption query with the $ID_A, ID_B, m$ as the input, $C$ firstly checks whether $ID_A = ID_B$. If $ID_A = ID_B$, $C$ executes the normal signcryption process according to our scheme. If $ID_A \neq ID_B$, $C$ queries the full private key $SK_B = (x_B, DB)$ of $ID_B$ and the full public key $PK_A = (P_A, RA)$ of $ID_A$ from $L_k$ and performs the following operation.

1. Chooses $r, v \in Z_q^*$ randomly and computes $R_i = r P - vR_A - vh_1(ID_A)P_{pub}$
2. Inserts into $L_3$ with the new item $(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R', h)$, where $h = v$
3. Computes $t_1 = R'D_B$ and $t_2 = P_A x_s$
4. Computes session key $k' = h_2(t_1, t_2, R_i)$, inserts into the $L_2$ with the new item $(t_1, t_2, R_i, k')$, and generates the ciphertext $c' = Enc_{k'}(m)$
5. Computes $s' = r/s$ and returns the ciphertext $\sigma' = (c', R', s')$ to $A_i$

After Bob receives the ciphertext $\sigma' = (c', R', s')$, it can pass the verification, with the following computation:

$$\hat{e}(R' + h'R_A + h'Q_A P_{pub}, P_{pub}) = \hat{e}(rP - vR_A - vQ_A P_{pub}) + vR_A + vQ_A P_{pub} + P_{pub}$$

$$= \hat{e}(rP, P_{pub}) \hat{e}(P, P_{pub})^{(r/s)} = P_A x_s$$

Unsigncryption query: when adversary $A_i$ executes the unsigncryption query with the $ID_A, ID_B, \sigma = (c, R, s)$ as the input, $C$ firstly checks whether $ID_B = ID_A$. If $ID_B \neq ID_A$, $C$ executes the normal unsigncryption process according to our scheme. If $ID_B = ID_A$, the full private key of $ID_B$ cannot be obtained, and $C$ queries $SK_A = (x_A, DA)$ of $ID_A$ and $PK_B = (P_B, RB)$ of $ID_B$ from $L_k$ and performs the following operation.

1. Computes $t_2 = (P_B)^{x_A}$
2. Retrieves each item $(t_1, t_2, R, k)$ in $L_2$ to determine whether $t_1 = RD_B$ is true. If the equation is true, it computes $k = h_3(t_1, t_2, R)$ and generates the message $m = Dec_c(c)$ or error symbol $\bot$
3. Gets $h$ from the list $L_3$ with $(ID_A, ID_B, P_A, P_B, R_A, P_{pub}, R, R_{pub})$ as input
4. Checks if the equation $P_A x' = \hat{e}(R + hR_A + h_1(ID_A) P_{pub}, P_{pub})$ holds, if true, $C$ returns the message $m$ to $A_i$; otherwise, $C$ returns the symbol $\bot$.

Challenge. After the queries in Phase 1, the adversary $A_i$ selects two same length plaintext $m_1, m_2$ and two user’s identity $ID_A, ID_B$. In this phase, $ID_A$ and $ID_B$ are the users that $A_i$ will challenge. Before the challenge, the full private key $SK_B$ cannot be queried and $PK_A$ cannot be replaced. If $ID_A \neq ID_B$ or $ID_B \neq ID_A$, $C$ aborts the game. Otherwise, $C$ generates the ciphertext $\sigma$ by following the steps.

1. Sets $R' = aP$, chooses $v \in Z_q^*$, $t'_1 \in G_1$, and computes $t_2 = P_B x_s$
2. Chooses session key $k' = h_3(t_1, t_2, R')$ and generates the ciphertext $c' = Enc_{k'}(m_1)$, where $\lambda \in \{0, 1\}$
3. Adds the new item $(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R', v)$ to the list $L_3$
4. Computes $s' = (a + h'^*DA)/x_A$, generates the ciphertext $\sigma^* = (c', R', s')$, and returns $\sigma^*$ to $A_i$.

Phase 2. In this stage, the adversary $A_i$ can execute the polynomially bounded queries like Phase 1. Note that $SK_B$ of $ID_B$ cannot be queried, $PK_A$ cannot be replaced, and the ciphertext $\sigma = (c, R, s)$ with sender $ID_B$ and receiver $ID_A$ cannot be executed unsigncryption query.

Guess. The challenger computes the probability of solving the CDH through the adversary $A_i$ that wins the game.
If the $A_1$ wins the game, the list $L_2$ will store $q_2$ items, one of which is the answer to the CDH problem. Thus, $C$ chooses the $t_1$ in every item of $L_2$ to test.

$$t_1 = \left( R_B + P_{pub} Q_B \right) r = (D_B) R = r D_B P = ab P$$

Analysis. Let us compute the probability of the challenger $C$ cab by solving the CDH problem. In this game, there are two conditions for the challenger to solve CDH: the challenger $C$ does not abort in this game and the adversary $A_1$ successfully wins this game. Firstly, the challenger $C$ abandons the challenge by following four events. Note that $q_T = q_1 + q_{sk} + q_{pk} + q_{pk}$.

1. Event 1: the adversary $A_1$ executes the partial private key query of $ID_J$, and the probability of this event is $q_{pk}/q_T$.
2. Event 2: the adversary $A_1$ executes the secret value query of $ID_J$, and the probability of this event is $q_{sk}/q_T$.
3. Event 3: the adversary $A_1$ executes the replace public key query of $ID_J$, and the probability of this event is $q_{pk}/q_T$.
4. Event 4: when the adversary $A_1$ executes challenges, the user $ID_B, ID_A$ are not challenged user identity, that is, $ID_B \neq ID_J, ID_A \neq ID_J$, and the probability of this event is $1 - 1/C_{q_1}^2$.

Therefore, we calculate the probability that $C$ does not give up the game with the probability $(1 - q_{pk}/q_T)(1 - q_{sk}/q_T)(1 - q_{pk}/q_T)(1/C_{q_1}^2)$. 

Secondly, the challenger $C$ goes through the list $L_2$ and selects $t_1$ as the answer to the CDH. Thus, the probability of finding the correct solution to CDH is $1/q_2$. Therefore, with the adversary $A_1$ as the subroutine, the challenger $C$ solves the CDH with the following probability $\varepsilon'$.

$$\varepsilon' = \varepsilon \left( 1 - q_{pk}/q_T \right) \left( 1 - q_{sk}/q_T \right) \left( 1 - q_{pk}/q_T \right) \left( \frac{1}{C_{q_1}^2} \right) \left( \frac{1}{q_2} \right)$$

Lemma 6. Assuming that the adversary $A_2$ can be able to win the game $Game_{A_2}^{IND-CCA2-2}$ with a nonnegligible advantages $\varepsilon$ after $q_1 h_1$ queries, $q_2 h_2$ queries, $q_3 h_3$ queries, $q_{pk}$ public key queries, and $q_{sk}$ secret value queries, the challenger $C$ can solve the CDH problem with the follow probability $\varepsilon'$.

$$\varepsilon' = \varepsilon \left( 1 - q_{sk}/q_T \right) \left( \frac{1}{C_{q_1}^2} \right) \left( \frac{1}{q_2} \right).$$

Proof. The challenger $C$ takes adversary $A_2$ as the subroutine, gives the adversary $A_2$ the $(P, P', P'')$. And the target is compute the $P_{ab}$ through the interactions between adversary $A_2$ and the challenger $C$. 

---

**Figure 1:** Comparison of running time for different message size.
Setup. The challenger $C$ executes the setup algorithm and generates the system public parameters $\{\}$.

Phase 1. The adversary $A_2$ executes the following polynomially bounded queries. Note that before the adversary executes queries, the challenger $C$ chooses randomly $I, J \in [1, q_1]$ for the challenged phase.

$h_1$ query: the challenger $C$ initializes and updates data table $L_1$, where the tuple format is $(ID_i, l_i)$. When adversary $A_2$ executes the $h_1$ query with the user identity $ID_i$ as input, $C$ firstly checks if $L_1$ contains the interrelated tuple $(ID_i, l_i)$. If $L_1$ contains the tuple $(ID_i, l_i)$, $C$ returns $l_i$ to $A_2$. If not, $C$ randomly chooses $l_i \in Z_q^*$, adds or updates the new item $(ID_i, l_i)$ to $L_1$, and returns $l_i$ to $A_2$.

$h_2$ query: the challenger $C$ initializes and updates data table $L_2$, where the tuple format is $(t_1, t_2, R, k)$. When adversary $A_2$ executes the $h_2$ query with the $(t_1, t_2, R, k)$ as the input, the challenger $C$ firstly checks if $L_2$ contains the interrelated tuple $(t_1, t_2, R, k)$. If $L_2$ contains the tuple $(t_1, t_2, R, k)$, $C$ returns the $k$ to $A_2$. If not, $C$ randomly chooses $k \in \{0, 1\}^*$, inserts the new item $(t_1, t_2, R, k)$ to $L_2$, and returns the $k$ to $A_2$.

$h_3$ query: the challenger $C$ initializes and updates data table $L_3$, where the tuple format is $(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R, h)$. When the adversary $A_2$ executes the $h_3$ query with the $(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R)$ as the input, the challenger $C$ firstly checks if $L_3$ contains the interrelated tuple. If $L_3$ contains this tuple, $C$ returns the $h$ to $A_2$. If not, $C$ randomly chooses $h \in Z_q^*$, adds or updates the new item $(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R, h)$ to $L_3$, and returns $h$ to $A_2$.

Secret value query: the challenger $C$ initializes and updates data table $L_4$, where the tuple format is $(ID_A, P_i, R_i, D_i, x_i)$. When adversary $A_2$ executes the secret value query with the $ID_A$ as the input, the challenger $C$ firstly checks if $L_4$ contains the interrelated item $(ID_A, x_i, \ldots, x_k)$. If $L_4$ contains this tuple, $C$ returns the $x_i$ to $A_2$. If not, $C$ randomly chooses $x_i \in Z_q^*$, computes $P_i = \hat{e}(P_{pub}, P)^{x_i}$, adds or updates the item $(ID_A, x_i, \ldots, x_k)$ to $L_4$, and returns $x_i$ to $A_2$. Note that if $ID_A = ID_1$, $C$ aborts this game.

Partial private key query: with the $ID_A$ as the input, the challenger $C$ firstly checks if $L_4$ contains the interrelated item $(ID_A, r_i, \ldots, r_k)$. If $L_4$ contains this tuple, $C$ returns the $R_i$ to $A_2$. Otherwise, $C$ randomly chooses $D_i \in Z_q^*$, adds or updates the item $(ID_A, r_i, \ldots, r_k)$ to $L_4$, and returns $D_i$ to $A_2$.

Public key query: with the $ID_A$ as the input, the challenger $C$ firstly checks if $L_4$ contains the interrelated item $(ID_A, P_i, R_i, \ldots, R_k)$. If $L_4$ contains this tuple, $C$ returns the $P_i$ to $A_2$. Otherwise, if $ID_A \neq ID_1$ or $ID_A \neq ID_J$, $C$ executes the secret value query with $ID_A$ gets the secret value $x_i$ from $L_4$, and computes $P_i = \hat{e}(P, P_{pub})^{x_i}$. If $ID_A = ID_1$, $C$ sets $a \in Z_q^*$ and computes $P_i = \hat{e}(P, P_{pub})^{x_i}$. If $ID_A = ID_J$, $C$ sets $b \in Z_q^*$ and computes $P_i = \hat{e}(P, P_{pub})^{x_i}$. Then, $C$ chooses $r_i \in Z_q^*$ and computes $R_i = r_i * P$, where $I_i$ is from $L_1$. Next, $C$ adds or updates the $L_4$ with the tuple $(ID_A, P_i, R_i, \ldots, R_k)$ and returns $PK_i = \{P_i, R_i\}$ to $A_2$.

Signcryption query: when adversary $A_2$ executes the signcryption query with the $(ID_A, ID_B, m)$ as the input, $C$ firstly checks whether $ID_A = ID_J$. If $ID_A \neq ID_J$, $C$ executes the normal signcryption process according to our scheme. If $ID_A = ID_J$, the full private key of $ID_A$ cannot be obtained, $C$ queries the full private key $SK_B = (x_B, D_B)$ of $ID_B$ and the full public key $PK_B = (P_B, R_B)$ of $ID_B$ from $L_k$ and performs the following operation.

1. Chooses $r, v \in Z_q^*$ and computes $R' = (r - v \hat{e}(A, v))P - vR_A + vR_A$.
2. Inserts into $L_3$ with the new item $(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R', h)$, where $h = v$.
3. Computes $t_1' = R'D_B$ and $t_2 = R_A^s$.
4. Computes session key $k' = h_2(t_1', t_2, R')$, inserts into $L_3$ with the new item $(t_1, t_2, R', k')$, and generates $c' = Enc_k(m)$.
5. Computes $s' = r/x_B$ and returns the ciphertext $\sigma' = (c', R', s')$ to $A_2$.

The adversary $A_2$ can pass the verification, with the following computation:

$$
\begin{align*}
\bar{e}(R' + h'R_A + h'R_B, P_{pub} - P_{pub}) & = \bar{e}(r - v \hat{e}(s))P - vR_A + vR_A + vQ_{AP}P_{pub}P_{pub}P_{pub}P_{pub}P_{pub}\,P_{pub}\,P_{pub}P_{pub}P_{pub}
\end{align*}
$$

Unsigncryption query: when adversary $A_2$ executes the unsigncryption query with the $(ID_A, ID_B, \sigma = (c, R, s))$ as the input, $C$ firstly checks whether $ID_A = ID_J$. If $ID_A \neq ID_J$, $C$ executes the normal unsigncryption process according to our scheme. If $ID_A = ID_J$, the full private key of $ID_A$ cannot be obtained, and $C$ queries $SK_A = (x_A, D_A)$ of $ID_A$ and $PK_B = (P_B, R_B)$ of $ID_B$ from $L_k$ and performs the following operation.

1. Computes $t_1 = RD_B$.
2. Retrieves each item $(t_1, t_2, R, k)$ in $L_2$ to determine whether $t_2 = (P_B)^{x_1}$. If the equation is true, it computes $k = h_1(t_1, t_2, R)$ and generates the message $m = Dec_k(c)$ or error symbol $\perp$.
3. Computes $h = h_3(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R)$.
4. Checks if the equation $P_A^t = \bar{e}(R + hR_A + hh_1(ID_A)P_{pub}, P_{pub})$ holds. If true, $C$ returns the message $m$ to $A_2$; otherwise, $C$ returns the symbol $\perp$ to $A_2$.

Challenge. After the queries in Phase 1, the adversary $A_2$ selects two same length plaintext $m_1, m_2$ and two user's
identity $ID_A, ID_B$. In this phase, $ID_A$ and $ID_B$ are the users that $A_2$ will challenge. Before the challenge, the secret value $x_B$ of $ID_B$ cannot be queried. If $ID_A \neq ID_I$ or $ID_B \neq ID_I$, the challenger $C$ aborts the game. Otherwise, $C$ generates the ciphertext $c$ by the following steps.

1. Chooses $r \in Z_q^*$, $v \in Z_q^*$, and $t_1^* \in G_2$. 
2. Computes $R^* = rP$ and $t_1 = r(R_B + P_{pub}Q_B)$.
3. Computes the session key $k^* = h_A(t_1, t_1^*, R^*)$ and generates the ciphertext $c^* = Enc_{c^*}(m_{\lambda})$, where $\lambda \in \{0, 1\}$.
4. Adds the new item $(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R^*, v)$ to the list $L_3$.
5. Computes $s^* = r + h^*D_A/x_A^*$, generates the ciphertext $\sigma^* = (c^*, R^*, s^*)$, and returns $\sigma^*$ to $A_2$.

Phase 2. In this stage, the adversary $A_2$ can execute the polynomially bounded queries like Phase 1. Note that the secret value $x_B$ of $ID_B$ cannot be queried and the ciphertext $\sigma = (c, R, s)$ with sender $ID_B$ and receiver $ID_J$ cannot be executed in the unsigncryption query.

Guess. The challenger computes the probability of solving the CDH through the adversary $A_2$ that wins the game. If the $A_2$ wins the game, the list $L_2$ will store $q_1$ items, one of which is the answer to the CDH problem. $C$ chooses the $t_2$ in every item for test.

$$t_2 = (P_B)^{s_1} = \tilde{v}(R, P_{pub})^{x_A s} = \tilde{v}(P, P_{pub})^{x_A s} = (P_A)^{x_A}$$ (9)

Analysis. Let us compute the probability of the challenger $C$ that can solve the CDH problem. In this game, there are two conditions for the challenger to solve CDH: the challenger $C$ does not abort in this game and the adversary $A_2$ successfully wins this game. Firstly, the challenger $C$ abandons the challenge by following two events. Note that $q_T = q_1 + q_{pk}$.

1. Event 1: the adversary $A_2$ executes the secret value query of $ID_J$, and the probability of this event is $q_{sk}/q_T$.
2. Event 2: when the adversary $A_2$ executes challenges, the user $ID_B, ID_A$ are not challenged user identity, that is, $ID_B \neq ID_I, ID_A \neq ID_I$. The probability of this event is $1 - 1/C_{q_I}^2$.

Therefore, we calculate the probability that $C$ does not give up the game with the probability $(1 - q_{sk}/q_T)(1/C_{q_I}^2)$. 

Secondly, the challenger $C$ goes through the list $L_2$ and selects $t_2$ as the answer to the CDH. Thus, the probability of finding the correct solution to CDH is $1/q_{T^1}$. Therefore, with the adversary $A_2$ as the subroutine, the challenger $C$ solves the CDH with the following probability $\epsilon'$.

$$\epsilon' = \epsilon \left(1 - \frac{q_{sk}}{q_T} \left(1 - \frac{1}{C_{q_I}^2} \frac{1}{q_{T^1}}\right)\right)$$ (10)

□

6.2. Unforgeability

Lemma 7. Assuming that the adversary $A_1$ can be able to win the game $\text{Game}^{EUF-CMA-1}$ with a nonnegligible advantages $\epsilon$ after $q_1h_1$ queries, $q_2h_2$ queries, $q_2h_3$ queries, $q_{pk}$ public key queries, $q_{sk}$ secret value queries, $q_{pk}$ the partial private key queries, and $q_{pk}$ the replace public key queries, the challenger $C$ can solve the DBDH problem with the follow probability $\epsilon'$.

$$\epsilon' = \epsilon \left(1 - \frac{q_{pk}}{q_T} \left(1 - \frac{q_{sk}}{q_T} \left(1 - \frac{q_{pk}}{q_T} \left(1 - \frac{1}{C_{q_I}^2}\right)\right)\right)\right).$$ (11)

Proof. The proof is similar to Lemma 5. The challenger $C$ takes adversary $A_1$ as the subroutine and gives the adversary $A_1$ the $(P, aP, bP, cP \in G_1, Z \in G_2)$. And the target is determined whether $P_{pub} = Z$ is true through the game between adversary $A_1$ and the challenger $C$. In Phase 1, the adversary $A_1$ can execute the polynomially bounded queries, and $C$ returns the corresponding answer to $A_1$. The adversary $A_1$ cannot perform the query about the full private key and replace public key of $ID_A$ and the signcryption query with the sender $ID_A$ and the receiver $ID_B$. In the forgery phase, the adversary $A_1$ generates the forgery ciphertext $\sigma' = (c', R', s')$. If the adversary $A_1$ wins the game and the challenger $C$ does not abort the game, $C$ makes the $a = x_A, b = r_x A, c = s$, and $Z = \tilde{v}(R, P_{pub})$. And the probability $\epsilon'$ that the challenger $C$ solves DBDH is the following. Note that $q_T = q_1 + q_{sk} + q_{pk} + q_{pk}$.

$$\epsilon' = \epsilon \left(1 - \frac{q_{pk}}{q_T} \left(1 - \frac{q_{sk}}{q_T} \left(1 - \frac{q_{pk}}{q_T} \left(1 - \frac{1}{C_{q_I}^2}\right)\right)\right)\right).$$ (12)

□

Lemma 8. Assuming that the adversary $A_2$ can be able to win the game $\text{Game}^{EUF-CMA-2}$ with a nonnegligible advantages $\epsilon$ after $q_1h_1$ queries, $q_2h_2$ queries, $q_2h_3$ queries, $q_{pk}$ public key queries, and $q_{sk}$ secret value queries, the challenger $C$ can solve the DDH problem with the follow probability $\epsilon'$. Note that $q_T = q_1 + q_{sk}$.

$$\epsilon' = \epsilon \left(1 - \frac{q_{sk}}{q_T} \left(\frac{1}{C_{q_I}^2}\right)\right).$$ (13)

Proof. The proof is similar to Lemma 6. The challenger $C$ takes adversary $A_1$ as the subroutine and gives the adversary
A2 the \((P^a, P^b, Z \in G_2)\). And the target is determined whether \(P^{ab} = Z\) is true through the interactions between adversary A1 and the challenger C. In Phase 1, the adversary A2 can execute the polynomially bounded queries, and C returns the corresponding answer to A2. The adversary A2 cannot perform the query about the secret value of IDA and the signcryption query with IDA to IDB. In the forgery phase, A2 generates the forgery ciphertext \(\sigma' = (c', R', s')\). If the adversary A2 wins the game and the challenger C does not abort the game, C makes the \(a = ss_A, b = hr_A/x_A\), and \(Z = \bar{v}(hR_A, P_{pub}) = P^{aR_x}\). And the probability \(\epsilon'\) that the challenger C solves DDH is the following:

\[
\epsilon' = \epsilon\left(1 - \frac{q_{ab}}{q_t}\right)\left(1 - \frac{1}{C^2_{qz}}\right). \tag{14}
\]

\(\square\)

6.3. Forward Security. To prove that our scheme meets forward security, we discuss the security situation of our scheme from two kinds of adversary A1, A2. For the adversary A1, it can obtain all the private key of the sender IDA and all the ciphertext information in the previous communication. However, it is difficult for the adversary A1 to decrypt the ciphertext and get the corresponding message m. The symmetric key of the ciphertext is generated by the hash function \(h_1\) with \(t_1, t_2\), and R. In the communication, the adversary A1 can recover \(t_1 = P^{aR_x}\) and get R from previous ciphertext. However, based on the CDH, the adversary A1 cannot calculate \(t_1 = RD_B(aP = R, bP = D_B)\). Therefore, adversary A1 cannot recover the message m from the previous communication. For the adversary A2, it can get the master system key s but cannot obtain the partial private key \(D_b\) of the receiver Bob. Thus, the proof of A2 is similar to that of the first type of attacker. Based on CDH, it is difficult for the adversary to calculate the \(t_1\) and the session key k and decrypt the ciphertext. Therefore, our scheme can meet forward security.

6.4. Public Verification. All users in the certificateless public key system can authenticate any ciphertext \(\sigma = (c, R, s)\). When receiving a ciphertext \(\sigma, ID_A, PK_A = (P_A, R_A), ID_B\) and \(PK_B = (P_B, R_B)\), any user of the system can compute \(h = h_3(ID_A, ID_B, P_A, P_B, R_A, R_B, P_{pub}, R)\) and execute the authentication equation \(P_A' = \bar{v}(R + hR_A + hQ_A, P_{pub})\) to determine whether the ciphertext is valid or not. In the process of authentication, decryption operations and the full private keys of the sender or receiver are not required, and verification can be quickly implemented. Therefore, our scheme can meet public verification.

6.5. Known Session-Specific Temporary Information Security. Assuming that the adversary has obtained the random value r of the sender’s signcryption process, the adversary still cannot obtain any valuable information. For the adversary A1, it can obtain \(ID_A, PK_A = (P_A, R_A), ID_B\), and \(PK_B = (P_B, R_B)\) and generates \(t_1 = (R_b + rQ_B)r\) and \(R = rP\). However, based the assumption of CDH problem, it is difficult for the adversary A1 to calculate the \(t_2 = P^{aR_x}\) and computes the correct session key k. For the adversary A2, it can obtain the master system key s of KGc but cannot obtain the secret value of \(x_A\). Thus, the proof of A2 is similar to the proof of A1 and the adversary A2 cannot correct session key k based on the assumption of CDH problem. Therefore, our scheme can meet known session-specific temporary information security.

7. Performance Analysis

7.1. Comparison of Communication Efficiency. In this section, we compared the communication efficiency of our scheme with other existed schemes and generate Table 2. In Table 2, m indicates the length of data to be transmitted, q indicates the length of each element in \(Z_{q^*}\), and \(g_i\) indicates the length of each element in \(G_1\).

Assume that we use the 80 bit security level in the standard model and the size of message m is 200 bits, the size of element in \(Z_{q^*}\) is 20 bits, and the size of element of \(G_1\) is 65 bits [21]. As can be seen from Table 2, in one communication, the message length of Luo et al.’s scheme is calculated \(l_{\text{in}} = 20 + 65 \times 2 + 200 = 350\) bits, the message length of Yin and Liang’s scheme is \(l_{\text{ypc}} = 20 \times 2 + 65 + 200 = 305\) bits, and the message length of our scheme is \(l_{\text{out}} = 20 + 65 + 200 = 285\) bits. Thus, compared with existing schemes, this scheme transmits less messages and has higher communication efficiency.

7.2. Comparison of Computational Efficiency. In this section, we analyze the computational efficiency of the proposed scheme through theoretical calculation and simulation experiments. In theoretical calculation, bilinear pairing computation, multiplication operation, and exponential operation are the three operations that consume the most computing resources in schemes. Therefore, we compare with other schemes on the number of operations of three kinds of calculations and generate Table 3, where mul, exp, and pair, respectively, stand for multiplication operation and exponential operation and bilinear pairing operation [22]. As can be seen from Table 3, compared with existing schemes, the proposed scheme reduces the calculation times of multiplication operation and bilinear pairing operation and reduces the calculation pressure of data encryption and decryption.

In the simulation experiment, we used ubuntu20.04 and pypbc to build a 512-bit standard security model. In the security model, we simulate the running process of the proposed scheme and the comparison schemes, measure the average running time under different size of message, and generate Figure 1. From Figure 1, in the process of data transmission, our scheme requires shorter running time than other schemes, and with the increase of the size of data transmission, our scheme has more obvious advantages in computing efficiency. Therefore, our proposed scheme reduces the computational pressure in the data encryption and decryption and is suitable for data transmission of the practical application of IoT.
8. Conclusion

In this paper, we firstly review and evaluate the certificateless hybrid signcryption proposed by Gong et al. [14]. It is pointed out that there are some deficiencies in its unforgeability. Based on the Gong et al.’s scheme, we propose an improved certificateless hybrid signcryption scheme for IoT and prove that the scheme can meet confidentiality, unforgeability, forward security, public verification, and known session-specific temporary information security in ROM. In addition, through theoretical analysis and simulation experiment, compared with the existing scheme, our scheme consumes less communication resources and computing resources, has higher communication efficiency and computing efficiency, and is suitable for data transmission of the practical application of IoT device.

In addition, it needs to be noted that this scheme increases the number of exponential operations in data transmission. In the future work, it can be considered to reduce the number of exponential operations, so as to further reduce the computing pressure of resource-constrained devices in IoT and improve the computing efficiency of this scheme.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Key Research and Development Project (grant number: 2019YFB2102300), National Natural Science Foundation of China (grant number: 61971014), and Young Backbone Teacher Training Program of Henan Colleges and Universities (grant number: 2021GGJS170).

References

[1] Z. A. Almusaylim and N. Zaman, “A review on smart home present state and challenges: linked to context-awareness Internet of Things (IoT),” Wireless Networks, vol. 25, no. 6, pp. 3193–3204, 2019.
[2] S. Lu, A. Wang, S. Jing et al., “A study on service-oriented smart medical systems combined with key algorithms in the IoT environment,” China Communications, vol. 16, no. 9, pp. 235–249, 2019.
[3] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, “Internet of Things (IoT) for next-generation smart systems: a review of current challenges, future trends and prospects for emerging 5G-IoT scenarios,” Access, vol. 8, pp. 23022–23040, 2020.
[4] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, “IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices,” IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8182–8201, 2019.
[5] C. Wheelus and X. Zhu, “IoT network security: threats, risks, and a data-driven defense framework,” IoT, vol. 1, no. 2, pp. 259–285, 2020.
[6] M. B. M. Noor and W. H. Hassan, “Current research on internet of things (IoT) security: a survey,” Computer Networks, vol. 148, pp. 283–294, 2019.
[7] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, “Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions,” Journal of Ambient Intelligence and Humanized Computing, pp. 1–18, 2017.
[8] S. Rajaee, V. Paul, V. G. Menon, and M. R. Khosravi, “A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices,” Symmetry, vol. 11, no. 2, p. 293, 2019.
[9] F. Li, M. Shirase, and T. Takagi, “Certificateless hybrid signcryption,” in International Conference on Information Security Practice and Experience, pp. 112–123, Springer, 2009.
[10] F.-T. Zhang, Y.-X. Sun, L. Zhang, M. Geng, and S. Li, “Research on certificateless public key cryptography,” Journal of Software, vol. 22, no. 6, pp. 1316–1332, 2011.
[11] H. Xiong, Y. Zhao, Y. Hou et al., “Heterogeneous signcryption with equality test for IoT environment,” IEEE Internet of Things Journal, vol. 8, no. 21, pp. 16142–16152, 2021.
[12] P.-Y. Ting, J.-L. Tsai, and T.-S. Wu, “Signcryption method suitable for low-power IoT devices in a wireless sensor network,” IEEE Systems Journal, vol. 12, no. 3, pp. 2385–2394, 2018.
[13] S. Yonezawa, T. Kobayashi, and T. Saito, Pairing-Friendly Curves, Network Working Group, Internet-Draft, 2019.
[14] B. Gong, Y. Wu, Q. Wang, Y.-H. Ren, and C. Guo, “A secure and lightweight certificateless hybrid signcryption scheme for Internet of Things,” Future Generation Computer Systems, vol. 127, pp. 23–30, 2022.
[15] A. Yin and H. Liang, “Certificateless hybrid signcryption scheme for secure communication of wireless sensor networks,” Wireless Personal Communications, vol. 80, no. 3, pp. 1049–1062, 2015.
[16] H.-F. Yu and B. Yang, “Low-computation certificateless hybrid signcryption scheme,” Frontiers of Information Technology & Electronic Engineering, vol. 18, no. 7, pp. 928–940, 2017.
[17] W. Luo and W. Ma, “Secure and efficient data sharing scheme based on certificateless hybrid signcryption for cloud storage,” Electronics, vol. 8, no. 5, p. 590, 2019.
[18] P. Kasoyoka, M. Kimwele, and S. M. Angolo, “Cryptanalysis of a pairing-free certificateless signcryption scheme,” ICT Express, vol. 7, no. 2, pp. 200–204, 2021.
[19] M. Luo, Y. Wan, and D. Huang, “Certificateless hybrid signcryption scheme with known session-specific temporary information security,” IF Network Security, vol. 19, no. 6, pp. 966–972, 2017.
[20] A. Yin and H. Liang, “On security of a certificateless hybrid signcryption scheme,” Wireless Personal Communications, vol. 85, no. 4, pp. 1727–1739, 2015.
[21] F. Li, Y. Han, and C. Jin, “Certificateless online/offline signcryption for the internet of things,” Wireless Networks, vol. 23, no. 1, pp. 145–158, 2017.
[22] B. Zhang, Z. Jia, and C. Zhao, “An efficient certificateless generalized signcryption scheme,” Security and Communication Networks, vol. 2018, 11 pages, 2018.